

Interval Superposition Arithmetic

Yanlin Zha · Boris Houska

Received: date / Accepted: date

Abstract This paper is about a novel set based computing method, called interval superposition arithmetic, for enclosing the image set of multivariate factorable functions on a given domain. In order to represent enclosure sets, the proposed arithmetic stores a matrix of intervals. Every point in the domain is associated with a sequence of interval valued components of this matrix and the superposition, i.e., the Minkowski sum, over these components is the actual enclosure of the function at this point. Interval superposition arithmetic has polynomial runtime complexity with respect to the number of variables of the factorable function. It is capable of representing highly complex enclosure sets, because the number of choices for picking a sequence of components from a given interval matrix grows exponentially with respect to the size of the matrix. The composition rules that are associated with interval superposition arithmetic exploit algebraic addition theorems of atom operations as well as partially separable sub-structures of the computational graph of factorable functions. Besides analyzing the accuracy and favorable convergence properties of interval superposition arithmetic, this paper illustrates the advantages of the proposed method compared to existing set arithmetics by studying numerical examples.

Keywords interval arithmetic · global optimization

1 Introduction

Tools for enclosing the image set of factorable functions are the basis for many reliable numerical computing methods including global optimization algorithms [15, 27], robust and semi-infinite optimization algorithms [14, 31], as well as validated integration algorithms [24, 44]. Here, factorable functions [29] are functions that can be represented as a finite recursive composition of atom operations from a (finite) library

$$\mathcal{L} = \{+, *, \sin, \exp, \log, \dots\}.$$

This library typically includes binary sums, binary products, and a number of univariate atom functions such as trigonometric functions, exponential functions, or logarithms. In

Yanlin Zha and Boris Houska are with the School of Information Science and Technology (SIST), ShanghaiTech University, 319 Yueyang Road, Shanghai 200031, China.
E-mail: borish@shanghaitech.edu.cn.

practice, factorable functions over a given library \mathcal{L} are represented in the form of a computational graph, which can be obtained conveniently in most object oriented programming languages by using operator overloading or source code transformation [16].

1.1 Review

Existing methods for computing enclosures of factorable functions can be divided into three categories: traditional interval arithmetics and its variants, arithmetics using other convex sets such as ellipsoids or zonotopes, as well as non-convex set arithmetics. In the following, advantages and limitations of these existing methods are reviewed.

Interval Arithmetic. Interval arithmetic is one of the oldest and most basic tools for computing enclosures of the image set of a factorable function on a given compact domain [33, 34]. Throughout this paper, we use the notation

$$\mathbb{I} = \{[a, b] \subseteq \mathbb{R} \mid a, b \in \mathbb{R}, a \leq b\}$$

to denote real valued intervals. Interval arithmetic proceeds by defining bounding rules for binary sums and products as well as for all univariate atom operations in the given library \mathcal{L} . For example, for two intervals $[a, b] \in \mathbb{I}$ and $[c, d] \in \mathbb{I}$, their sum, product, and exponential are given by

$$[a, b] + [c, d] = [a + b, c + d], \quad (1)$$

$$[a, b] * [c, d] = [\min\{ac, ad, bc, bd\}, \max\{ac, ad, bc, bd\}], \quad (2)$$

$$\text{and } e^{[a, b]} = [e^a, e^b]. \quad (3)$$

The derivation of other univariate composition rules is straightforward for most commonly used univariate atom operations [34]. Additionally, in order to ensure that an interval arithmetic is compatible with the addition and multiplication with scalars $c \in \mathbb{R}$, the notation $c + A = A + c$ with $A = [a, b]$ is used to denote the shifted interval $[c + a, c + b]$. Similarly, $cA = Ac$ denotes the scaled interval $[ca, cb]$ if $c \geq 0$ and $[cb, ca]$ if $c < 0$. Unfortunately, one of the main limitations of interval arithmetics is that the computed interval enclosures are often much wider than the exact range of the given factorable function. This overestimation effect is mainly caused by the so called *dependency problem*. The dependency problem can already be observed for very simple functions, such as $f(x) = x - x = x + (-1) * x$, $x \in [0, 1]$. Here, the exact image set is $[0, 0]$, but a naive application of standard interval arithmetics gives the interval bound $[-1, 1]$. On the other hand, an advantage of interval arithmetics is its favorable computational complexity: storing an interval in floating point format amounts to storing two floating point numbers of the same format. In terms of computation time, evaluating a product of intervals is about 4-5 times more expensive than evaluating a product of real numbers. The above rules for the binary sum and evaluation of the exponential functions are only 2-times more expensive than a corresponding function evaluation with real numbers. Another advantage of interval arithmetics is that the computational and storage complexity of its multivariate extension scales linearly with the number of variables of the factorable function. This is due to the fact that the complexity of storing an n -dimensional interval vector $x \in \mathbb{I}^n$ is of order $\mathbf{O}(n)$.

Convex Set Arithmetics. During the last decades, there have been many attempts to overcome the limitations of interval arithmetics by using more advanced set propagation techniques. Here, one way to generalize interval arithmetics is to replace the intervals (or interval vectors) with more general computer representable convex sets. For example, McCormick relaxations propagate convex lower and concave upper bounds rather than standard intervals [29, 32]. McCormick’s arithmetic has the same computational complexity as interval arithmetics with respect to the number of variables n , namely $\mathbf{O}(n)$. In practice, it sometimes yields tighter bounds, but it is also slightly more expensive than interval arithmetics despite the fact that it has the same complexity, as illustrated in [32]. Another class of convex set based enclosure tools is the so-called ellipsoidal calculus [21, 43], where multi-dimensional ellipsoids rather than interval vectors are used in order to represent the set enclosures. In terms of storage requirements, n -dimensional ellipsoids can be represented by a vector in \mathbb{R}^n , representing the center, and a symmetric and positive semi-definite shape matrix in $\mathbb{R}^{n \times n}$. Thus, the complexity of storing ellipsoids is of order $\mathbf{O}(n^2)$, while intervals can be stored with complexity $\mathbf{O}(n)$. Unfortunately, the computational complexity of ellipsoidal calculus is typically of order $\mathbf{O}(n^3)$, if dense matrix-matrix multiplications are not avoided. Concerning applications in the field of validated integration and control, ellipsoidal calculus has been found to be a suitable tool for mitigating the inherent dependency problem of set arithmetics [19]. For example, in contrast to interval arithmetics, ellipsoid arithmetics can be used to construct stable validated integration algorithms [20]. In this sense, it has been established that, at least for particular applications, the higher computational effort of storing ellipsoids rather than intervals pays out in terms of the accuracy of the enclosure set. Other convex enclosure methods use polyhedral sets, which are in general even more expensive to store than ellipsoids, but they can be used to represent convex sets with arbitrary precisions by controlling the number of facets. Polyhedral relaxations have become popular in the field of global optimization and are for example used in the software BARON [38, 40], which is based on linear programming relaxations. Another example for an enclosure algorithms based on polyhedral sets is the so-called “affine arithmetics” [13]. Affine arithmetic is based on zonotopes, a particular class of point-symmetric polytopes.

Non-Convex Set Arithmetics. A rather apparent disadvantage of all convex set based arithmetics is that they can, in the best case, represent the convex hull of the image of a given factorable function. Consequently, if the exact image set of a factorable function is non-convex, the benefit of investing into more accurate convex set representations, such as zonotopes or even general polytopes with many facets, is limited. One way to overcome this limitation is by working with non-convex sets, which are in practice often represented by using polynomials. The use of interval polynomials or polynomials with interval remainder terms have been developed since a long time starting in the 1960s [33] and 1980s [12, 37]. These early works have been the basis for the popular Taylor model arithmetic, which has been developed by Berz and coworkers [4, 5, 28]. Nowadays, there exist mature tools, for example the software MC++ [32], implementing Taylor model arithmetics with arbitrary order. The favorable convergence properties of Taylor models on variable domains with small diameter have been analyzed thoroughly [6]. However, the convergence properties of Taylor series on wider domains are often less favorable. For example, the Taylor series of the function $f(x) = (1 + x^2)^{-1}$ at $x = 0$ is divergent outside the interval $[-1, 1]$, as the analytic continuation of this function has poles at $\pm i$, $i = \sqrt{-1}$. A promising direction towards overcoming this limitation of Taylor models is the ongoing research on so-called Chebychev models. For functions with one or two variables Chebychev models can be constructed by the software **Chebfun** as developed by Trefethen and coworkers [3, 41, 42]. Chebychev models for func-

tions with more than two variables are the focus of recent research [36]. While computing bounds on convex sets is computationally tractable, finding tight bounds of a multivariate polynomial is itself a complex task. Here, one way to compute bounds on such polynomials is to use linear matrix inequalities [22]. Other heuristics for computing range bounds for multivariate polynomials can be found in [23].

1.2 Motivation

A principal goal of this paper is to develop a non-convex set arithmetic for factorable function that exploits *global algebraic structures* rather than attempting to borrow methods from local and numerical analysis such as variational analysis, Taylor expansions, or other polynomial approximation techniques. The following paragraphs outline two global algebraic structures that are exploited by the proposed interval superposition arithmetic, namely, *addition theorems* [1, 2, 45] and *additive separability* [9, 10].

Addition Theorems. A rational addition theorem for a univariate atom operation $g : D \rightarrow \mathbb{R}$ with domain $D \subseteq \mathbb{R}$ is a formula that expresses $g(x+y)$ as a rational function of $g(x)$ and $g(y)$,

$$g(x+y) = \mathcal{R}_g(g(x), g(y)) \quad \text{for all } x, y \in D \text{ with } x+y \in D \quad (4)$$

with $\mathcal{R}_g : \mathbb{R} \times \mathbb{R} \rightarrow \overline{\mathbb{R}} = \mathbb{R} \cup \{\infty\}$ being a bivariate rational function. An important example is the addition theorem for the exponential function, $e^{x+y} = e^x e^y$, which holds globally on the domain $D = \mathbb{R}$. Notice that the right-hand side of the addition theorem for the sine function, $\sin(x+y) = \sin(x)\cos(y) + \cos(x)\sin(y)$, is an exception in the sense that its right-hand side depends not only $\sin(x)$ and $\sin(y)$ but also $\cos(x)$ and $\cos(y)$. However, if we allow complex arguments, this addition theorem can alternatively be regarded as a special case of the addition theorem for the exponential function such that it can be written in the form of (4). A similar statement holds for the cosine function. Moreover, in analogy to the univariate case, a rational addition theorem for a bivariate atom operation $g : D \rightarrow \mathbb{R}$ with domain $D = D_1 \times D_2$, $D_1, D_2 \subseteq \mathbb{R}$ is a formula of the form

$$g(a+b, c+d) = \mathcal{R}_g(g(a, c), g(a, d), g(b, c), g(b, d))$$

for all $a, b \in D_1$ with $a+b \in D_1$ and all $c, d \in D_2$ with $c+d \in D_2$. Here, $\mathcal{R}_g : \mathbb{R} \rightarrow \mathbb{R}$ is a rational function with 4 arguments. An incomplete list of examples for addition theorems, which are either in this form or in generalized versions relevant for the derivations in this paper, are collected in Table 1. An important property of addition theorems is that they hold globally on the whole domain of a function. Also notice that many addition theorems contain other important algebraic relations as a special case. For example, the addition theorem for the product contains the binomial formulas

$$(x+y)^2 = x^2 + 2xy + y^2 \quad \text{and} \quad (x+y)(x-y) = x^2 - y^2$$

as special cases. Moreover, substituting $y = -x$ in the addition theorem of the cosine function yields the equation $\sin(x)^2 + \cos(x)^2 = 1$. Similarly, the addition theorem for the exponential contains the equation $e^{-x} = \frac{1}{e^x}$ as a special case. Also these equations hold globally, for all x . A recursive application of the above addition theorems can be used to derive formulas for more general sums, in the context of the present paper called *superpositions*, of more than

Domain	Atom Operation	Addition Theorem
$\mathbb{R} \times \mathbb{R}$	sum	$(a + b) + (c + d) = (a + c) + (b + d)$
\mathbb{R}	unary minus	$-(x + y) = (-x) + (-y)$
$\mathbb{R} \times \mathbb{R}$	product	$(a + b)(c + d) = ac + bc + ad + bd$
\mathbb{R}_{++}	inverse	$(x + y)^{-1} = \frac{x^{-1}y^{-1}}{x^{-1} + y^{-1}}$
\mathbb{R}	exponential	$e^{x+y} = e^x e^y$
\mathbb{R}	sine	$\sin(x + y) = \sin(x)\cos(y) + \cos(x)\sin(y)$
\mathbb{R}	cosine	$\cos(x + y) = \cos(x)\cos(y) - \sin(x)\sin(y)$
$(-\frac{\pi}{2}, \frac{\pi}{2})$	tangent	$\tan(x + y) = \frac{\tan(x) + \tan(y)}{1 - \tan(x)\tan(y)}$
$(0, \pi)$	cotangent	$\cot(x + y) = \frac{\cot(x)\cot(y) - 1}{\cot(x) + \cot(y)}$

Table 1 A list of rational addition theorems for common atom operations. The addition theorems for the sine and cosine functions can alternatively be regarded as special cases of the addition theorem for the exponential function, if complex arguments are allowed.

two summands, e.g., in order to express $g(x + y + z)$ in terms of $g(x)$, $g(y)$, and $g(z)$. Also notice that some atom operations admit inverse addition theorems, which express the sum of $g(x)$ and $g(y)$ by evaluating g at a rational expression that depends on x and y . An important example for an inverse addition theorem is the formula

$$\log(x) + \log(y) = \log(xy) \quad \text{for } x, y > 0, \quad (5)$$

which can be obtained by inverting the addition theorem for the exponential function. Other inverse addition theorems can be obtained by starting with an addition theorem and rewriting it in terms of the corresponding inverse atom operation, g^{-1} , although the corresponding formulas can become more involved. For example, the formula

$$\arctan(x) + \arctan(y) = \arctan\left(\frac{x+y}{1-xy}\right) \pmod{\pi}, \quad xy \neq 1 \quad (6)$$

has to be read modulo π and also holds for $xy \neq 1$ only. An important observation is that many but not all practically relevant atom operations admit either rational addition theorems or inverse rational addition theorems. At this point it should be mentioned that most articles on addition theorems can be found in the complex analysis literature, which usually analyze algebraic rather than rational addition theorems [1, 2, 45]. Algebraic addition theorems are more general than rational additional theorems, as they allow the case that $g(x + y)$, $g(x)$, and $g(y)$ are the roots of a multivariate polynomial. For example, the square-root function satisfies the algebraic equation

$$\forall x, y \geq 0, \quad (\sqrt{x+y})^2 - (\sqrt{x})^2 - (\sqrt{y})^2 = 0,$$

which is an algebraic addition theorem, but $\sqrt{\cdot}$ does not admit a rational addition theorem, as it is not possible to express $\sqrt{x+y}$ as a rational function of \sqrt{x} and \sqrt{y} .¹ However, in

¹ If it were possible to write $\sqrt{x+y}$ as a rational expression of \sqrt{x} and \sqrt{y} , we could substitute $x = y = 1$ to show that $\sqrt{2}$ is a rational number, which leads to a contradiction.

this paper the motivation to introduce addition theorems arises from algorithmic considerations rather than from a complex analysis context. From the perspective of computational algorithm development it is desirable to avoid implicit equations whenever possible and, therefore, as much as this paper borrows the concept of addition theorems from complex analysis, it does not consider functions like the square-root as independent atom operations. In the case of square-roots and related algebraic operations this is justified by the fact that we can use globally valid formulas like

$$\forall x > 0, \quad \sqrt{x} = e^{\frac{1}{2} \log(x)}$$

to express these functions as compositions of admissible atom operations for which rational addition theorems or inverse rational addition theorems are available. In summary, this paper envisions a set arithmetics that benefits from exploiting both rational addition theorems and their associated inverse forms.

Complete Additive Separability. For any given univariate function $f : X \rightarrow \mathbb{R}$ on a given interval $X \subseteq \mathbb{R}$ one way to improve the accuracy of a set arithmetic, e.g., traditional interval arithmetic, is to combine it with branching. Here, the main idea is to divide the interval $X = [\underline{x}, \bar{x}]$ into N sub-intervals and apply interval calculus to bound f on each of these sub-intervals separately. Unfortunately, such an exhaustive branching approach is prohibitively expensive in higher dimensional spaces. For example, if every coordinate of a multi-dimensional interval box $X \in \mathbb{I}^n$ is divided into N equidistant pieces, the function f has to be bounded on N^n sub-intervals separately, a prohibitively large number if $n, N \gg 1$. This problem is known under the name *curse of dimensionality*. However, one important class of functions f for which this exhaustive branching approach is not affected by the curse of dimensionality are the so-called (completely) separable functions $f : X \rightarrow \mathbb{R}$ with interval domain $X = X_1 \times X_2 \times \dots \times X_n \in \mathbb{I}^n$. Here, a function $f : X_1 \times X_2 \times \dots \times X_n \rightarrow \mathbb{R}$, $X_i \in \mathbb{I}$, is called (completely) separable, if it can be written in the form of a superposition,

$$f(x) = \sum_{i=1}^n \phi_i(x_i),$$

of given factorable functions $\phi_i : X_i \rightarrow \mathbb{R}$, $i \in \{1, \dots, n\}$. The global minimizer (maximizer) of a first order separable function f can be found by minimizing (maximizing) the summands ϕ_i separately. If the functions ϕ_i are Lipschitz continuous, we can compute upper and lower bounds of f with accuracy $\mathbf{O}\left(\frac{n}{N}\right)$ within a polynomial run-time of order $\mathbf{O}(nN)$ by dividing each of the intervals X_i into N sub-intervals and using standard interval calculus to bound the ϕ_i s [34]. Practically relevant functions are often not separable themselves, but parts of their computational graph may be separable and it is desirable to develop set arithmetics, which can exploit such structure.

Partial Additive Separability. The concept of partial additive separability has originally been introduced by Conn, Gould, and Toint [9, 10] for analyzing the structure of large scale optimization problems. A factorable function $f : X \rightarrow \mathbb{R}$, $X \subseteq \mathbb{R}^n$, is called partially separable, if there exist m matrices $U_1 \in \{0, 1\}^{n_1 \times n}, \dots, U_m \in \{0, 1\}^{n_m \times n}$ such that every of the $n_i \leq n$ rows of U_i contains exactly one 1 and such that f can be written in the form

$$\forall x \in X, \quad f(x) = \sum_{i=1}^m \phi_i(U_i x).$$

Here, the functions $\phi_i : X_i \rightarrow \mathbb{R}$, $X_i = U_i X \subseteq \mathbb{R}^{n_i}$, are assumed to be factorable. Throughout this paper, the *coupling degree* of a partially separable factorable function f is defined to be the maximum number of rows of the matrices U_i , denoted by

$$\deg(f) = \max_{i \in \{1, \dots, m\}} n_i .$$

Notice that factorable functions with coupling degree 1 are completely separable. This definition is based on the assumption that factorable functions are regarded as symbolic objects, which are defined by their computational graph rather than by their mapping properties. For example, the second order polynomials

$$f(x) = (x_1 + x_2 + x_3)^2 \quad (7)$$

$$\text{and } g(x) = x_1^2 + 2x_1x_2 + x_2^2 + 2x_2x_3 + x_3^2 + 2x_1x_3 \quad (8)$$

are equivalent in the sense that they have the same mapping behavior and image sets, because the addition theorem for products implies that we have $f(x) = g(x)$ for all $x \in \mathbb{R}^3$. However, the functions f and g are not considered equal in this paper, $f \neq g$, as the computational graphs of f and g do not coincide. In particular, the coupling degree of the factorable function f is 3, while the coupling degree of the factorable function g is 2. The above example can be generalized: for every multivariate polynomial of degree q there exists an equivalent factorable function with coupling degree of at most q , which can be obtained by applying addition theorems if necessary. Unfortunately, although global extrema of a factorable function with coupling degree 1 can be found in polynomial run-time, an analogous statement does not hold true in general for factorable functions with coupling degree ≥ 2 . For example, nonconvex quadratic programming problems are, in general, NP-hard [35]. However, there exist efficient semi-definite programming relaxations for bounding the minimizer and maximizer of nonconvex quadratic forms over an interval domain, which have originally been analyzed in the context of the Maximum Cut problem [17]. More general sub-optimality estimates of semi-definite programming relaxations can be found in [18, 25, 26]. Moreover, the global optimization solver GLoMIO [30] focusses on nonconvex quadratically constrained quadratic programming problems. All these global optimization tools specialize on classes of optimization problems whose objective and constraint functions are factorable functions with a small coupling degree, namely a coupling degree of 2. One principal motivation of this paper is to systematically detect and exploit the presence of such partially separable structures within the computational graph of generic factorable functions by a suitable set arithmetic.

1.3 Contribution

The main contribution of this paper is the development of a novel interval superposition arithmetic for enclosing the image set of factorable functions on a given interval domain. As far as the authors are aware, the presented set arithmetic is the first enclosure method that exploits the availability of rational addition theorems as well as inverse rational additional theorems of its underlying library of atom operations for the construction of set-valued models and their remainder bounds. Another key contribution of the proposed enclosure arithmetic is that it is, on the one hand, solely based on polynomial run-time algorithms, but, on the other hand, systematically uses exhaustive (facet-aligned) branching of the whole domain into smaller pieces in order to mitigate the conservatism of the computed enclosure

sets. The authors argue that this way of constructing set superpositions for the implementation of enclosure tools might lead to new types of global optimization algorithms, where the traditional “Branch” and “Bound” operations are not implemented as separate routines anymore. The proposed superposition arithmetic considers branching as a subroutine of the enclosure arithmetic (bounding), which leads to a framework that can enforce polynomial run-time and storage requirements on the maintained partition, while the conservatism of the enclosures can be controlled if partially separable structures are present. The paper is organized as follows.

Section 2 introduces first order interval superposition arithmetics in order explain the main ideas of constructing and using superposition models (Sections 2.1 and 2.2). This section is written in tutorial style and introduces the reader to the main conceptual ideas before discussing the more powerful but also more advanced high-order generalizations in Section 3, which are based on more abstract notation. Relevant contributions of this section are Algorithm 1 and Algorithm 2 in Sections 2.3 and 2.4, respectively, which introduce univariate and bivariate composition rules for first order superposition arithmetics. Here, rational addition theorems as well as inverse rational addition theorems are used for the construction of remainder bounds, which are derived in all detail in Appendix A. Moreover, in Section 2.6 it is shown that first order interval superposition have a locally linear Hausdorff convergence order, which tends to become quadratic if the branching accuracy is increased. Therefore the local behavior is similar to first order Taylor methods or centered forms in standard interval arithmetic. However, the main difference to these existing enclosure methods is that first order interval superposition models have favorable global accuracy and convergence properties on wider domains if additional separability structures are present. The corresponding analysis results can be found in Section 2.7.

Section 3 proposes a non-trivial higher-order generalization of interval superposition arithmetics thereby taking an important step towards turning the developed enclosure tools into a set arithmetic with globally controllable accuracy on arbitrary domains. Although this section contains some of the results from Section 2 as a special case, the authors recommend to start reading Section 2 before proceeding with the more advances developments in Section 3. In detail, the proposed techniques are based on algebraic dependency expansions, which can be interpreted as a derivative-free algebraic abstraction of Taylor expansions as elaborated in Section 3.1. These dependency expansions turn out to be a powerful basis for the derivation of general univariate and bivariate compositions rules as introduced in Sections 3.2 and 3.3. Moreover, on a more technical level, Sections 3.4 and 3.5 discuss how to automate the derivation of associated remainder bounds for higher order superposition arithmetics by using rational addition theorems as well as inverse rational addition theorems featuring modern computer algebra tools. Finally, Section 3.6 summarizes the complexity of the proposed superposition arithmetic in dependence on the expansion order. It is shown that the proposed set enclosure method has polynomial storage and run-time complexity for any given dependency expansion order.

Section 4 presents numerical results that are based on a prototype implementation of the proposed interval superposition arithmetic, which is written in the programming language JULIA. In detail, Section 4.1 illustrates the advantages of the proposed set arithmetic compared to existing tools such as Taylor model based enclosure methods. Moreover, Section 4.2 analyzes a second case study by introducing a factorable function whose evaluation is based

on a multivariate vector-valued discrete-time recursion. This leads to a case study with challenging dependency structure, which is used to illustrate the advantages of second order interval superposition models compared to their first order variant. The numerical results for both case studies highlight the main practical contribution of the proposed interval superposition arithmetics, namely, that this arithmetic can be used to compute accurate bounds on factorable functions on wider domains, where other methods, such as Taylor models, are divergent or very conservative.

Finally, Section 5 concludes the paper.

2 First Order Interval Superposition Arithmetic

2.1 Interval Superposition Models

Let $f : X \rightarrow \mathbb{R}$ be a given factorable function and $X \subseteq \mathbb{R}^n$ a given interval domain, which is assumed to have the form

$$X = [\underline{x}_1, \bar{x}_1] \times [\underline{x}_2, \bar{x}_2] \times \dots [\underline{x}_n, \bar{x}_n] \in \mathbb{I}^n.$$

A set valued function $F_{f,X} : X \rightarrow \mathbb{I}$ is called an interval valued enclosure function of the given factorable function f on the given domain X , if it satisfies

$$\forall x \in X, \quad f(x) \in F_{f,X}(x).$$

The goal of this and the next sections is to develop polynomial run-time algorithms, which can construct piecewise constant enclosure functions $F_{f,X}$ in such a way that the approximation error, i.e., the maximum width of the intervals $F_{f,X}(x)$ on X , denoted by

$$\text{diam}(F_{f,X}(x)) = \max_{y_1, y_2 \in F_{f,X}(x)} |y_1 - y_2|,$$

is small. In the following, coordinate aligned branching is applied in order to cut the whole domain into smaller intervals of the form

$$X_i^j = [\underline{x}_i + (j-1)h_i, \underline{x}_i + jh_i] \quad \text{with} \quad h_i = \frac{\bar{x}_i - \underline{x}_i}{N} \quad (9)$$

for all $i \in \{1, \dots, n\}$ and all $j \in \{1, \dots, N\}$, where N is an integer that the user can choose. Here, the intervals $[\underline{x}_i, \bar{x}_i]$ are all cut into N equidistant intervals for simplicity of presentation, although the following algorithms can easily be generalized for non-equidistant interval branching and for the case that each coordinate is not necessarily subdivided into the same number N of sub-intervals. Notice that dividing every coordinate into N subintervals amounts to dividing the whole domain into N^n subintervals. Clearly, an application of traditional interval arithmetics or any other of the above reviewed existing set arithmetics to each of these N^n intervals separately is prohibitively expensive in high dimensional spaces. Such exhaustive approaches would be in conflict with the principal goal of this paper to develop polynomial run-time enclosure algorithms. Thus, in order to avoid the curse of dimensionality, the basis functions

$$\phi_i^j(x) = \begin{cases} 1 & \text{if } x_i \in X_i^j \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

for all $i \in \{1, \dots, n\}$ and all $j \in \{1, \dots, N\}$ are introduced. Next, the goal is to develop an arithmetic that computes piecewise constant enclosure functions of the form

$$F_{f,X}(x) = \sum_{i=1}^n \sum_{j=1}^N A_i^j \phi_i^j(x), \quad (11)$$

where the coefficients $A_i^j \in \mathbb{I}$ are intervals. The enclosure function in (11) is called a *first order interval superposition model*. This name is motivated by the fact that $F_{f,X}(x)$ is represented as a Minkowski sum of n interval valued functions. Notice that the complexity of storing a first order interval superposition model is $2nN$, as we need to store the upper and lower bounds of the nN intervals A_i^j . The function $F_{f,X}(x)$ is piecewise constant in x and may take different interval values on all of its N^n pieces while the complexity of storing $F_{f,X}(x)$ is only $2nN$. Figure 1 visualizes this aspect for a two-dimensional example. For first order superposition models we call the index i in (11) the row index of the coefficient matrix

$$A = \begin{pmatrix} A_1^1 & \dots & A_1^N \\ \vdots & \ddots & \vdots \\ A_n^1 & \dots & A_n^N \end{pmatrix}$$

Similarly, j is called the column index. Notice that this matrix notation is only introduced in order to have a convenient storage format for the interval coefficients. Moreover, for the case that the coordinate aligned intervals are not divided into the same number of pieces the rows of the storage matrix have different individual lengths.

Remark 1 Notice that there is more than one way to represent the same interval superposition model. This is mainly due to the fact that the enclosure set

$$F_{f,X}(x) = \sum_{i=1}^n \sum_{j=1}^N A_i^j \phi_i^j(x), \quad (12)$$

remains invariant if we pick two pairwise disjoint row indices, $k_1 \neq k_2$, and a constant $c \in \mathbb{R}$, add the offset c to all intervals in the k_1 -th row, and subtract c from all intervals in the k_2 -th row,

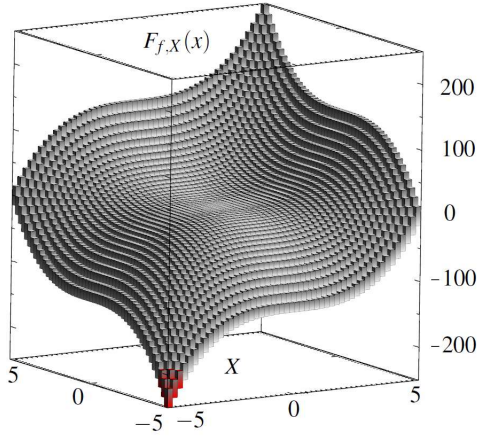
$$\forall j \in \{1, \dots, N\}, \quad A_{k_1}^j \leftarrow A_{k_1}^j + c \quad \text{and} \quad A_{k_2}^j \leftarrow A_{k_2}^j - c.$$

In principle, such redundancies could be removed by using a sparse interval matrix A , which maintains systematically as many zero interval entries as possible. However, for the theoretical derivations in this paper such a redundancy free storage scheme is not convenient, as it leads to cumbersome notation. Therefore the following theoretical sections of this paper proceed without imposing additional sparsity requirements on the coefficient matrix A .

2.2 Range Bounders

If $F_{f,X}$ is a first order interval superposition enclosure of f , bounds of this function can be found by computing the global minimum and global maximum of the enclosure, given by

$$\lambda(A) / \mu(A) = \min_{x,y} / \max_{x,y} y \quad \text{s.t.} \quad \begin{cases} y \in F_{f,X}(x) \\ x \in X \end{cases}$$



Example 1. A first order interval superposition enclosure $F_{f,X}(x)$ for the separable function

$$f(x) = x_1^3 + x_2^3$$

for $N = 50$ on the interval domain

$$X = [-5, 5] \times [-5, 5].$$

The function $F_{f,X}$ is a union of 2500 intervals, but only $2 \times 50 = 100$ intervals are stored. Candidates for the global minimizer of f on X are colored in red.

Fig. 1 Visualization of a first order interval superposition model.

The functions λ and μ are called range boundaries. In order to develop an algorithmic procedure for evaluating these functions for a given interval matrix A , the following notation for the row-wise upper and lower bounds is introduced:

$$U(A_i) = \max_{j \in \{1, \dots, N\}} \bar{A}_i^j \quad \text{and} \quad L(A_i) = \min_{j \in \{1, \dots, N\}} \underline{A}_i^j \quad \text{with} \quad A_i^j = [\underline{A}_i^j, \bar{A}_i^j].$$

The exact range boundaries of $F_{f,X}$ can now be evaluated by using the following result.

Proposition 1 *The exact range boundaries of the enclosure function $F_{f,X}$ are given by*

$$\lambda(A) = \sum_{i=1}^n L(A_i) \quad \text{and} \quad \mu(A) = \sum_{i=1}^n U(A_i).$$

Proof. The main idea is to exploit complete separability of first order interval superpositions models,

$$\forall x \in X, \quad F_{f,X}(x) = \sum_{i=1}^n \underbrace{\left[\sum_{j=1}^N A_i^j \phi_i^j(x) \right]}_{\text{depends on } x_i \text{ only}}. \quad (13)$$

The definition of the basis functions ϕ_i^j in (10) implies that $\phi_i^j(x)$ depends on the i -th component of x only, i.e., the summands in the above expression can be minimized and maximized separately finding the componentwise extrema $L(A_i)$ and $U(A_i)$, respectively. The sum of these extrema corresponds to the exact range bounder of $F_{f,X}$, as stated by the proposition. \square

An immediate consequence of the above proposition is that if $F_{f,X}$ is an enclosure function of f on X , then upper and lower bounds on the function f on the domain X are given by

$$\forall x \in X, \quad \lambda(A) = \sum_{i=1}^n L(A_i) \leq f(x) \leq \sum_{i=1}^n U(A_i) = \mu(A).$$

Notice that the cost of evaluating the functions U and L for one row A_i is of order $\mathbf{O}(N)$. Thus, if $F_{f,X}(x)$ is a given superposition model of f , the cost of computing the above upper and lower bounds $\mu(A)$ and $\lambda(A)$ is of order $\mathbf{O}(nN)$, as the functions U and L have to be evaluated for all n rows of the coefficient matrix A and added up.

2.3 Univariate Compositions

This section focusses on the question how to propagate first order interval superposition models through univariate compositions with atom functions. Let

$$F_{h,X}(x) = \sum_{i=1}^n \sum_{j=1}^N A_i^j \phi_i^j(x)$$

denote a first order interval superposition model of the function $h : X \rightarrow \mathbb{R}$ and let $g : \mathbb{R} \rightarrow \mathbb{R}$ denote a given univariate atom function from the atom library \mathcal{L} . As it was already outlined in the introduction, examples for common univariate atom functions are exp, log, sin, cos, etc.. The goal of this section is to find a first order interval superposition model

$$F_{f,X}(x) = \sum_{i=1}^n \sum_{j=1}^N C_i^j \phi_i^j(x) \quad \text{of the function} \quad f = g \circ h.$$

Here, the notation $g \circ h$ denotes the composition of g and h , i.e., such that $(g \circ h)(x) = g(h(x))$ for all x . The input of a composition rule for a first order interval superposition arithmetic for a univariate atom operation $g \in \mathcal{L}$ are the coefficients A_i^j and its output are the coefficients C_i^j such that whenever $F_{h,X}(x)$ is an enclosure function of h on X , then $F_{f,X}$ is an enclosure function of $f = g \circ h$ on X . The construction of a valid map from A to C depends on the particular form of the univariate atom function g , but, fortunately, the main concept for computing C is shared by all univariate composition rules. Thus, the focus of the following discussion is on Algorithm 1, which outlines this main concept for the systematic construction of composition rules. Algorithm 1 has a computational complexity of order $\mathbf{O}(nN)$. Moreover, the validity of the computed coefficients C_i^j is established by the following theorem.

Theorem 1 *If $F_{h,X}(x) = \sum_{i=1}^n \sum_{j=1}^N A_i^j \phi_i^j(x)$ is an enclosure function of h on X and if the coefficient C_i^j are computed by Algorithm 1, then the function*

$$F_{f,X}(x) = \sum_{i=1}^n \sum_{j=1}^N C_i^j \phi_i^j(x)$$

is an enclosure function of $f = g \circ h$.

Proof. Let $x \in X$ be any point in the interval X . Since $F_{h,X} = \sum_{i=1}^n \sum_{j=1}^N A_i^j \phi_i^j(x)$ is a first order interval superposition enclosure of the function h , there must exist a sequence of integers $j_1, j_2, \dots, j_n \in \{1, \dots, N\}$ and associated points $y_i \in A_i^{j_i}$ such that

$$h(x) = \sum_{i=1}^n y_i.$$

Algorithm 1: Composition rule of first order interval superposition arithmetic

Input: Interval valued coefficients $A_i^j \in \mathbb{I}$ of the input enclosure model $F_{h,X}$ and an atom function $g \in \mathcal{L}$.

Main Steps:

1. Choose for all $i \in \{1, \dots, n\}$ suitable central points $a_i \in \mathbb{R}$ such that

$$L(A_i) \leq a_i \leq U(A_i) \quad \text{and set} \quad \omega = \sum_{i=1}^n a_i.$$

2. Choose a suitable remainder bound $r_g(A) \geq 0$ such that

$$\left| \sum_{i=1}^n g(\omega + \delta_i) - (n-1)g(\omega) - g\left(\omega + \sum_{i=1}^n \delta_i\right) \right| \leq r_g(A) \quad (14)$$

for all $\delta \in \mathbb{R}^n$ with $\forall i \in \{1, \dots, n\}, L(A_i) \leq a_i + \delta_i \leq U(A_i)$.

3. Compute the interval valued coefficients

$$C_i^j = g\left(\omega - a_i + A_i^j\right) - \frac{n-1}{n}g(\omega).$$

for all $i \in \{1, \dots, n\}$ and all $j \in \{1, \dots, N\}$, where $g\left(\omega - a_i + A_i^j\right)$ is evaluated by using traditional interval arithmetic.

4. Pick a suitable $k \in \{1, \dots, n\}$ and set $C_k^j \leftarrow C_k^j + r_g(A) \cdot [-1, 1]$ for all $j \in \{1, \dots, N\}$.

Output: The coefficients C_i^j of a first order interval superposition model $F_{f,X}$ of the function $f = g \circ h$.

Next, we define $\delta_i = y_i - a_i$ and recall the definition $\omega = \sum_{i=1}^n a_i$ from Step 1 of Algorithm 1. These definitions can be used to write the function $f(x)$ in the form

$$\begin{aligned} f(x) &= g(h(x)) = g\left(\sum_{i=1}^n y_i\right) = g\left(\omega + \sum_{i=1}^n \delta_i\right) \\ &= \sum_{i=1}^n \left(g(\omega + \delta_i) - \frac{n-1}{n}g(\omega) \right) - \underbrace{\left(\sum_{i=1}^n g(\omega + \delta_i) - (n-1)g(\omega) - g\left(\omega + \sum_{i=1}^n \delta_i\right) \right)}_{\in r_g(A) \cdot [-1, 1]}. \end{aligned}$$

As we have $\delta_i \in A_i^{j_i} - a_i$, the inclusion $g(\omega + \delta_i) \in g(\omega - a_i + A_i^{j_i})$ holds. Consequently, we have

$$f(x) \in \sum_{i=1}^n \left(g(\omega - a_i + A_i^{j_i}) - \frac{n-1}{n}g(\omega) \right) + r_g(A) \cdot [-1, 1] = \sum_{i=1}^n C_i^{j_i}.$$

This implies that $F_{f,X}(x) = \sum_{i=1}^n \sum_{j=1}^N C_i^j \phi_i^j(x)$ is a first order interval superposition enclosure of the function $f = g \circ h$. \square

The most important step of Algorithm 1 is Step 2, where a remainder bound $r_g(A)$ has to be constructed. This remainder bound is required to satisfy the inequality

$$\left| \sum_{i=1}^n g(\omega + \delta_i) - (n-1)g(\omega) - g\left(\omega + \sum_{i=1}^n \delta_i\right) \right| \leq r_g(A) \quad (15)$$

for all $\delta \in \mathbb{R}^n$ with $L(A_i) \leq a_i + \delta_i \leq U(A_i)$ for all $i \in \{1, \dots, n\}$. As it has been discussed in the introduction, many commonly used atom operations admit either rational addition theorems or inverse rational addition theorems. Thus, although the construction of r_g depends on the particular form of the atom function $g \in \mathcal{L}$, two main strategies for constructing this bound can be outlined.

Strategy 1. If the atom operation g admits a rational addition theorem, this theorem can be used in order to express the terms $g(\omega + \delta_i)$ and $g(\omega + \sum_{i=1}^n \delta_i)$ as rational functions of $g(\delta_1), \dots, g(\delta_n)$ and $g(\omega)$. Consequently, there exists a rational function \mathcal{R}_g^n such that²

$$\sum_{i=1}^n g(\omega + \delta_i) - (n-1)g(\omega) - g\left(\omega + \sum_{i=1}^n \delta_i\right) = \mathcal{R}_g^n(g(\delta_1), \dots, g(\delta_n), g(\omega)) . \quad (16)$$

Fortunately, bounds on $g(\delta_1), \dots, g(\delta_n)$ and $g(\omega)$ can be found easily by using traditional interval arithmetic or any other existing set arithmetic. Based on these bounds, the outer rational expression \mathcal{R}_g^n can then be processed further in a systematic way by using customized symbolic transformations in order to bound its image set. Numerous examples on how this procedure can be automated for particular instances of commonly used atom functions g and their addition theorems can be found in Appendix A.

Strategy 2. If the function g admits an inverse rational addition theorem, the associated formula can be used to express the left hand side of (15) by evaluating the function g at a rational expression of $\delta_1, \dots, \delta_n$ and ω . In other words, there exists a rational function \mathcal{R}_g^n such that

$$\sum_{i=1}^n g(\omega + \delta_i) - (n-1)g(\omega) - g\left(\omega + \sum_{i=1}^n \delta_i\right) = g(\mathcal{R}_g^n(\delta_1, \dots, \delta_n, \omega)) . \quad (17)$$

This expression can be bounded by first constructing bounds on the inner rational expression $\mathcal{R}_g^n(\delta_1, \dots, \delta_n, \omega)$ in combination with traditional interval arithmetic or other bounding tools to enclose the outer composition with g . Examples on how to apply this strategy for particular atom operations g and their inverse rational addition theorems can be found in Appendix A, too.

Notice that for both strategies the central points a_i can be regarded as additional degrees of freedom for minimizing the size of the remainder bound $r_g(A)$, as these variables can be chosen freely in Step 1 of Algorithm 1. Central points and remainder bounds for particular atom operations, which can be found by applying the above mentioned strategies, are listed in Table 2 and the details of the corresponding derivations can be found in Appendix A. The first column of Table 2 specifies the domain of the atom function while the second column specifies the atom function itself. If the domain is \mathbb{R}_{++} , we have to assert $\lambda(A) > 0$. If this assertion is violated the remainder bound is set to $r_g(A) = \infty$. For example, for the function $g(x) = x^{-1}$ with domain \mathbb{R}_{++} we have to assert $\lambda(A) > 0$, as we cannot exclude division by zero otherwise. Similarly, for the function $g(x) = \tan(x)$, we have to assert $-\frac{\pi}{2} < \lambda(A)$ and $\mu(A) < \frac{\pi}{2}$. The third column of Table 2 contains an expression for the central points a_i (for all $i \in \{1, \dots, n\}$) as needed in Step 1 of Algorithm 1. Finally, the fourth column

² Recall that the addition theorem for the sine and cosine function can be recovered as the real and imaginary part of complex valued rational addition theorem $e^{i(x+y)} = e^{ix}e^{iy}$, $i = \sqrt{-1}$, such that (16) can be used for this case, too.

Domain	$g(x)$	Central points	Remainder bound
\mathbb{R}	$-x$	$a_i = \frac{U(A_i)+L(A_i)}{2}$	$r_g(A) = 0$
\mathbb{R}	x^2	$a_i = \frac{U(A_i)+L(A_i)}{2}$	$r_g(A) = \sum_{i=1}^n (\sigma - s_i) s_i$ with $s_i = \frac{U(A_i)-L(A_i)}{2}$ and $\sigma = \sum_{i=1}^n s_i$
\mathbb{R}_{++}	x^{-1}	$a_i = \frac{L(A_i)\mu(A)}{\lambda(A)+\mu(A)} + \frac{U(A_i)\lambda(A)}{\lambda(A)+\mu(A)}$	$r_g(A) = \frac{\sum_{i=1}^n s_i (\mu(A) - \omega - (U(A_i) - a_i))}{\omega \lambda(A)}$ with $s_i = \max \left\{ \frac{a_i - L(A_i)}{\omega - a_i + L(A_i)}, \frac{U(A_i) - a_i}{\omega - a_i + U(A_i)} \right\}$
\mathbb{R}	e^x	$a_i = \log \left(\frac{e^{U(A_i)} + e^{L(A_i)}}{2} \right)$	$r_g(A) = e^\omega [\prod_{i=1}^n (1 + s_i) - \sum_{i=1}^n s_i - 1]$ with $s_i = \frac{e^{U(A_i)} - e^{L(A_i)}}{e^{L(A_i)} + e^{U(A_i)}}$
\mathbb{R}_{++}	$\log(x)$	$a_i = \frac{U(A_i)+L(A_i)}{2}$	$r_g(A) = -\log \left(1 - \frac{\prod_{i=1}^n (\omega + s_i) - \omega^{n-1} (\omega + \sum_{i=1}^n s_i)}{\omega^{n-1} \lambda(A)} \right)$ with $s_i = \frac{U(A_i)-L(A_i)}{2}$
\mathbb{R}	$\sin(x)$	$a_i = \frac{U(A_i)+L(A_i)}{2}$	$r_g(A) = \Omega (\prod_{k=1}^n (1 + s_k) - \sum_{k=1}^n s_k - 1)$ with $\Omega = \sin(\omega) + \cos(\omega) $, $s_i = 2 \sin([-r_i, r_i]) $, and $r_i = \frac{U(A_i)-L(A_i)}{4}$
\mathbb{R}	$\cos(x)$	same as for $\sin(x)$	same as for $\sin(x)$
$(-\frac{\pi}{2}, \frac{\pi}{2})$	$\tan(x)$	$a_i = \frac{U(A_i)+L(A_i)}{2}$	$r_g(A) = \sum_{i=1}^{n-1} \tan(S_{i+1}) \tan(\sum_{k=1}^i S_k) \tan(\sum_{k=1}^{i+1} S_k) $ $* [1 + \tan(\omega) \tan(\omega + \Sigma)]$ $+ \sum_{i=1}^n \tan(\omega) \tan(S_i) \tan(T_i)$ $* [1 + \tan(\omega + S_i) \tan(T_i) \tan(\omega + \Sigma)]$ with $s_i = \frac{U(A_i)-L(A_i)}{2}$, $S_i = [-s_i, s_i]$, $\sigma = \sum_{i=1}^n s_i$ and $\Sigma = [-\sigma, \sigma]$, $T_i = [-\sigma + s_i, \sigma - s_i]$

Table 2 Central points and remainder bounds for common univariate atom functions as needed in Step 1 and Step 2 of the composition rule of the first order interval superposition arithmetic (Algorithm 1).

contains an expression for an associated remainder bound that satisfies (14). Notice that the implementation of the interval superposition arithmetics tools that are presented in this paper avoid to duplicate code whenever possible. For example, the remainder terms for the sine and cosine function have the same form. Similarly, a separate implementation of the function $g(x) = x^{-1}$ for negative x is not needed, as we can combine the atom operations $g(x) = x^{-1}$ and $g(x) = -x$ in order to avoid redundancies. Moreover, the cotangent function can be written in the form $\cot(x) = \tan(\frac{\pi}{2} - x)$ such that an independent implementation is not necessary.

Remark 2 As discussed in Remark 1 the proposed interval superposition model storage scheme is redundant with respect to offset shifts. Consequently, in Step 4 of Algorithm 1

the remainder can in principle be added to any row of the matrix C . One possible implementation heuristic is to add the remainder to a row, which contains the intervals with the maximum average diameter. One could also imagine a more advanced implementation, which stores a sparse interval matrix A and then maintains as many zero entries as possible in order to avoid redundancies. Sparse interval superposition models are, however, beyond the scope of the current paper.

2.4 Bivariate Compositions

This section discusses how to construct arithmetic rules for first order interval superpositions for bivariate operators. Here, we can restrict ourself to the addition and multiplication, as for two given factorable functions h and g their difference and quotient,

$$h(x) - g(x) = h(x) + (-g(x)) \quad (18)$$

$$\text{and } h(x)/g(x) = h(x) * (1/g(x)) , \quad (19)$$

can be obtained by combining binary addition and multiplication with univariate mirroring and univariate inversion. Fortunately, the addition of two given interval superposition models is straightforward: if

$$F_{h,X}(x) = \sum_{i=1}^n \sum_{j=1}^N A_i^j \phi_i^j(x) \quad \text{and} \quad F_{g,X}(x) = \sum_{i=1}^n \sum_{j=1}^N B_i^j \phi_i^j(x)$$

are first order interval superposition enclosures of the given functions $h, g : X \rightarrow \mathbb{R}$, then

$$F_{f,X}(x) = \sum_{i=1}^n \sum_{j=1}^N C_i^j \phi_i^j(x) \quad \text{with} \quad C_i^j = A_i^j + B_i^j$$

is an enclosure of the function $f(x) = h(x) + g(x)$. Algorithm 2 outlines a product rule. Similar to Algorithm 1, the complexity of Algorithm 2 is of order $\mathbf{O}(nN)$. The validity of the bounds from Algorithm 2 is established in the following theorem.

Theorem 2 *If the $F_{h,X}(x) = \sum_{i=1}^n \sum_{j=1}^N A_i^j \phi_i^j(x)$ and $F_{g,X}(x) = \sum_{i=1}^n \sum_{j=1}^N B_i^j \phi_i^j(x)$ are first order interval superposition models of the functions $h, g : X \rightarrow \mathbb{R}$ and if the coefficient C_i^j is computed by Algorithm 1, then the function*

$$F_{f,X}(x) = \sum_{i=1}^n \sum_{j=1}^N C_i^j \phi_i^j(x)$$

*is a first order interval superposition enclosure of the function $f = h * g$.*

Proof. Let $x \in X$ be any point in the interval X . Since $F_{h,X}$ and $F_{g,X}$ are first order interval superposition enclosures of the functions h and g , there must exist a sequence of integers $j_1, j_2, \dots, j_n \in \{1, \dots, N\}$ and associated points $y_i \in A_i^{j_i}$ as well as $z_i \in B_i^{j_i}$ such that

$$h(x) = \sum_{i=1}^n y_i \quad \text{and} \quad g(x) = \sum_{i=1}^n z_i .$$

Algorithm 2: Product Rule of First Order Interval Superposition Arithmetic

Input: Interval valued coefficients $A_i^j \in \mathbb{I}$ and $B_i^j \in \mathbb{I}$ of the factors.

Main Steps:

1. Compute the central points

$$\forall i \in \{1, \dots, n\}, \quad a_i = \frac{U(A_i) + L(A_i)}{2} \quad \text{and} \quad b_i = \frac{U(B_i) + L(B_i)}{2}$$

and set

$$\alpha = \sum_{i=1}^n a_i, \quad \beta = \sum_{i=1}^n b_i, \quad \text{and} \quad \gamma = \sum_{i=1}^n a_i b_i \quad \text{as well as} \quad \omega = \frac{1}{n} [\alpha\beta - \gamma].$$

2. Compute the row-wise radii

$$\rho_i(A) = \frac{U(A_i) - L(A_i)}{2} \quad \text{and} \quad \rho_i(B) = \frac{U(B_i) - L(B_i)}{2}$$

for all $i \in \{1, \dots, n\}$ as well as the associated remainder bound

$$R(A, B) = \left(\sum_{i=1}^n \rho_i(A) \right) \left(\sum_{i=1}^n \rho_i(B) \right) - \sum_{i=1}^n \rho_i(A) \rho_i(B).$$

3. Compute the output coefficients

$$C_i^j = \left(A_i^j + \alpha - a_i \right) \left(B_i^j + \beta - b_i \right) - (\alpha - a_i)(\beta - b_i) - \omega$$

for all $i \in \{1, \dots, n\}$ and all $j \in \{1, \dots, N\}$, where the product $A_i^j B_i^j$ is evaluated by using the traditional interval arithmetic product rule from (2).

4. Pick a suitable $k \in \{1, \dots, n\}$ and set $C_k^j \leftarrow C_k^j + R(A, B) \cdot [-1, 1]$ for all $j \in \{1, \dots, N\}$.

Output: The coefficients C_i^j of a first order interval superposition model that encloses the product of the input models.

Next, we use the equation

$$\begin{aligned} f(x) &= h(x) * g(x) = \left(\sum_{i=1}^n y_i \right) * \left(\sum_{i=1}^n z_i \right) \\ &= \sum_{i=1}^n [y_i z_i + y_i(\beta - b_i) + (\alpha - a_i)z_i - \omega] - \left(\sum_{i=1}^n (y_i - a_i) \right) \left(\sum_{i=1}^n (z_i - b_i) \right) \\ &\quad + \sum_{i=1}^n (y_i - a_i)(z_i - b_i). \end{aligned} \tag{20}$$

Here, the latter equation follows from the addition theorem for the product rule in combination with a substitution of the equations

$$\alpha = \sum_{i=1}^n a_i, \quad \beta = \sum_{i=1}^n b_i, \quad \text{and} \quad \omega = \frac{1}{n} \left[\left(\sum_{i=1}^n a_i \right) \left(\sum_{i=1}^n b_i \right) - \sum_{i=1}^n a_i b_i \right]$$

in order to collect terms. The construction of the remainder bound $R(A, B)$ in Step 2 of Algorithm 2 is such that we have

$$\left| \left(\sum_{i=1}^n (y_i - a_i) \right) \left(\sum_{i=1}^n (z_i - b_i) \right) - \sum_{i=1}^n (y_i - a_i)(z_i - b_i) \right| \leq R(A, B)$$

for all $y_i \in A_i^{j_i}$ and all $z_i \in B_i^{j_i}$. Consequently, we have

$$f(x) \in \sum_{i=1}^n \left(A_i^{j_i} B_i^{j_i} + A_i^{j_i} (\beta - b_i) + (\alpha - a_i) B_i^{j_i} - \omega \right) + R(A, B) \cdot [-1, 1] = \sum_{i=1}^n C_i^{j_i}.$$

This implies that $F_{f,X}(x) = \sum_{i=1}^n \sum_{j=1}^N C_i^j \phi_i^j(x)$ is a first order interval superposition enclosure of the function $f = h * g$. \square

Notice that Algorithm 2 has many similarities with Algorithm 1. In particular, Algorithm 2 has the same run-time complexity $\mathbf{O}(nN)$.

2.5 Initialization

Algorithm 1 and 2 can be combined in order to implement the proposed first order interval superposition arithmetic by either operator overloading or source code transformation. This is in complete analogy to the implementation of traditional interval arithmetic or other set propagation methods following the directed acyclic computational graph of the given factorable function. The corresponding procedure is initialized by constructing (trivial) first order interval superposition models of all input variables x_i . As x_i does not depend on other variables its associated interval coefficients $A_k^j = 0$ can be set to 0 for all $k \neq i$ and all $j \in \{1, \dots, N\}$. The remaining i -th row of the interval coefficient matrix is initialized by

$$\forall j \in \{1, \dots, N\}, \quad A_i^j = X_i^j,$$

recalling that the branches X_i^j have been defined in (9).

2.6 Local Convergence Analysis

The proposed first interval superposition arithmetics is affected by two sources of overestimation. The first source of overestimation comes from the fact that scalar functions, such as $f(x) = x$, can be represented by first order interval superposition models with finite accuracy only. However, for Lipschitz continuous functions this error is of order $\mathbf{O}(\frac{1}{N})$ and can thus be controlled by choosing N sufficiently large. Therefore, the focus of the following analysis is on the second source of overestimation, namely the remainder bounds $r_g(A)$ and $R(A, B)$, which are needed in Algorithm 1 and Algorithm 2. The following lemma analyzes the local properties of the term that must be bounded by $r_g(A)$.

Lemma 1 *If the function $g : \mathbb{R} \rightarrow \mathbb{R}$ is twice continuously differentiable, then*

$$\left| \sum_{i=1}^n g(\omega + \delta_i) - (n-1)g(\omega) - g\left(\omega + \sum_{i=1}^n \delta_i\right) \right| \leq \mathbf{O}\left([\mu(A) - \lambda(A)]^2\right).$$

for all $\delta \in \mathbb{R}^n$ with $|\delta_i| \leq U(A_i) - L(A_i)$.

Proof. Let g' denote the derivative of the function g . As g is twice continuously differentiable, we can substitute the Taylor expansions

$$\sum_{i=1}^n g(\omega + \delta_i) = ng(\omega) + g'(\omega) \sum_{i=1}^n \delta_i + \mathbf{O}\left(\sum_{i=1}^n \delta_i^2\right)$$

as well as

$$g\left(\omega + \sum_{i=1}^n \delta_i\right) = g(\omega) + g'(\omega) \sum_{i=1}^n \delta_i + \mathbf{O}\left(\left[\sum_{i=1}^n \delta_i\right]^2\right).$$

Consequently, we have

$$\left| \sum_{i=1}^n g(\omega + \delta_i) - (n-1)g(\omega) - g\left(\omega + \sum_{i=1}^n \delta_i\right) \right| \leq \mathbf{O}\left(\left[\sum_{i=1}^n \delta_i\right]^2\right). \quad (21)$$

Next, we can use the inequality $|\delta_i| \leq U(A_i) - L(A_i)$ in combination with the triangle inequality and Proposition 1 to find

$$\left| \sum_{i=1}^n \delta_i \right| \leq \sum_{i=1}^n |\delta_i| \leq \sum_{i=1}^n (U(A_i) - L(A_i)) = \mu(A) - \lambda(A). \quad (22)$$

The statement of the lemma follows now by combining the inequalities (21) and (22). Notice that this result is independent of the choice of the central points a_i . \square

The above lemma motivates that a reasonable requirement on the remainder bound function $r_g : \mathbb{I}^{n \times N} \rightarrow \mathbb{R}$ from Algorithm 1 is that it satisfies

$$\forall A \subseteq \overline{D}, \quad r_g(A) \leq \mathbf{O}\left([\mu(A) - \lambda(A)]^2\right), \quad (23)$$

where $\overline{D} \subseteq D$ is a compact subset of the (open) domain D of the atom function D . It can be checked easily that this requirement is indeed satisfied all remainder bounds that are listed in Table 2. The details of this statement can be found in Appendix A.

Lemma 2 *The remainder term $R(A, B)$ of Algorithm 1 satisfies*

$$R(A, B) \leq \frac{1}{4}(\mu(A) - \lambda(A))(\mu(B) - \lambda(B)).$$

Proof. The definition of $R(A, B)$ in Step 2 of Algorithm 2 is such that the inequality

$$R(A, B) \leq \left(\sum_{i=1}^n \rho_i(A)\right) \left(\sum_{i=1}^n \rho_i(B)\right) = \frac{1}{4}(\mu(A) - \lambda(A))(\mu(B) - \lambda(B)) \quad (24)$$

holds, as stated by the lemma. \square

By combining the results of Lemma 1 and 2 a simple induction argument implies that the local convergence properties of first order interval superposition arithmetic can be summarized as follows.

Theorem 3 *Let all atom operations $g \in \mathcal{L}$ be twice continuously differentiable and let the remainder bounds r_g of all univariate atom operations satisfy (23). The maximum distance between the upper and lower bound of a first order interval superposition model $F_{f,X}$ computed by the above outlined arithmetic rules satisfies*

$$\max_{x \in X} \{ \text{diam}(F_{f,X}(x)) \} \leq \mathbf{O} \left(\frac{\text{diam}(X)}{N} + [\text{diam}(X)]^2 \right),$$

for all intervals $X \subseteq \overline{D}$, where $\overline{D} \subset D$ is a compact subset of an open domain D on which the function f has no singularities. Here, $\text{diam}(X)$ denotes the diameter of the interval domain X .

Proof. The statement of this theorem follows from the fact that variables can be represented with accuracy $\mathbf{O} \left(\frac{\text{diam}(X)}{N} \right)$ (induction start) while the remainder bound contributions from each atom operation can be bounded by expressions of order $\mathbf{O} \left(\frac{\text{diam}(X)}{N} + [\text{diam}(X)]^2 \right)$ by using the results from Lemma 1 and 2 (induction step). The details of this induction argument are straightforward and skipped for the sake of brevity. \square

At this point, one might argue that the convergence rate of first order interval superposition arithmetic is only linear with respect to the diameter of X . However, first of all, the constant in front of the linear term scales with $\frac{1}{N}$ and can thus be made arbitrarily small by choosing a sufficiently large N . And secondly, one possible path towards generalizing the above superposition arithmetic could be to construct a superposition of Taylor models or other sets rather than intervals, if the goal is to move towards better local properties. However, recall that the focus of the proposed arithmetic is *not* on the local properties for the case that X has a very small diameter and therefore we do not analyze this generalization any further. Rather, the goal of this paper is to develop an arithmetic, which has favorable properties on wider domains by exploiting global algebraic structures.

Remark 3 The statement of Theorem 3 may fail to hold for intervals domains that converge to a singularity of f . In order illustrate this problem we analyze the function

$$f(x) = \frac{1}{x_1 + x_2}$$

on the interval $X = [\varepsilon, 2\varepsilon] \times [\varepsilon, 2\varepsilon] \subseteq \mathbb{R}^2$ for a small $\varepsilon > 0$. Notice that the diameter of the X converges to 0 for $\varepsilon \rightarrow 0$, but the overestimation error of the proposed enclosure arithmetic is divergent. More precisely, if we apply the proposed procedure by using the remainder bound estimate for the inverse operation in Table 2, we find that the corresponding remainder bound satisfies

$$r_{\text{Interval Sup.}} = \frac{1}{20\varepsilon},$$

which diverges for $\varepsilon \rightarrow 0$. On the one hand, one might argue that this is much worse than standard interval arithmetic, which yields exact bounds for this particular example. On the other hand, first order Taylor models with interval remainders would yield a remainder bound with radius

$$r_{\text{Taylor}} = \frac{1}{8\varepsilon},$$

which, in turn, is worse than the remainder bound obtained with the proposed first order interval superposition arithmetic. However, the superior performance of standard interval

arithmetics for this particular example is rather accidental and due to the single occurrence of the variables x_1 and x_2 in the evaluation tree of the function f . In fact, numerical examples that illustrate the advantages of the proposed superposition arithmetic for more involved examples compared to both standard interval arithmetics and Taylor models can be found in Section 4.

2.7 Global Properties of First Order Interval Superposition Arithmetic

The main motivation for the development of interval superposition arithmetics is that it has favorable global properties. In order to discuss this the following definition of the coupling degree of an interval superposition model is useful.

Definition 1 A first order interval superposition model $F_{f,X}(x) = \sum_{i=1}^n \sum_{j=1}^N A_i^j \phi_i^j(x)$ is said to have degree 1, if there exist an integer $k \in \{1, \dots, n\}$ such that

$$L(A_i) = U(A_i) \quad \text{for all } i \in \{1, \dots, n\} \setminus \{k\}.$$

An immediate consequence of the initialization routine from Section 2.5 is that the interval superposition model of every variable has degree 1. For the univariate composition rule the following result can be established.

Lemma 3 Let the interval superposition model $F_{h,X}(x) = \sum_{i=1}^n \sum_{j=1}^N A_i^j \phi_i^j(x)$ of the inner function $h(x)$ of the composition rule have degree 1 in the sense of Definition 1, then we have

$$\left| \sum_{i=1}^n g(\omega + \delta_i) - (n-1)g(\omega) - g\left(\omega + \sum_{i=1}^n \delta_i\right) \right| = 0$$

for all $\delta \in \mathbb{R}^n$ with $L(A_i) \leq a_i + \delta_i \leq U(A_i)$ for all $i \in \{1, \dots, n\}$.

Proof. If $F_{h,X}$ is an interval superposition model with degree 1, we must choose $L(A_i) = a_i = U(A_i)$ for all indices $i \in \{1, \dots, n\} \setminus \{k\}$ for a fixed $k \in \{1, \dots, n\}$. Thus, $\delta_i = 0$ is the only possible choice for all $i \neq k$, which implies

$$\left| \sum_{i=1}^n g(\omega + \delta_i) - (n-1)g(\omega) - g\left(\omega + \sum_{i=1}^n \delta_i\right) \right| = |g(\omega + \delta_k) - g(\omega + \delta_k)| = 0.$$

This is the statement of the lemma. \square

The above lemma implies that the remainder bound function $r_g : \mathbb{I}^{n \times N} \rightarrow \mathbb{R}$ can be constructed in a way that $r_g(A) = 0$ whenever the input model $F_{f,X}(x) = \sum_{i=1}^n \sum_{j=1}^N A_i^j \phi_i^j(x)$ has degree 1. It can be checked easily that all remainder bounds from Table 2 have this property.

Lemma 4 If the input models of the product rule from Algorithm 2 have both degree 1 with respect to the same index k , i.e., if there exists an integer $k \in \{1, \dots, n\}$ such that

$$\forall i \in \{1, \dots, n\} \setminus \{k\}, \quad L(A_i) = U(A_i) \quad \text{and} \quad L(B_i) = U(B_i), \quad (25)$$

then the remainder term $R_j(A, B)$ of Algorithm 2 satisfies $R(A, B) = 0$.

Proof. If the input models satisfy condition (25), then the equation

$$\forall i \in \{1, \dots, n\} \setminus \{k\}, \quad \rho_i(A) = \rho_i(B) = 0$$

is satisfied. A substitution of this equation in the definition of R from Step 2 of Algorithm 2 yields

$$\begin{aligned} R(A, B) &= \left(\sum_{i=1}^n \rho_i(A) \right) \left(\sum_{i=1}^n \rho_i(B) \right) - \sum_{i=1}^n \rho_i(A) \rho_i(B) \\ &= \rho_k(A) * \rho_k(B) - \rho_k(A) * \rho_k(B) = 0. \end{aligned}$$

This is the statement of the lemma. \square

A combination of the above lemmata yields the following global statement about the accuracy of the proposed first order interval superposition arithmetic.

Theorem 4 *If f is a (completely) separable function and factorable over the library \mathcal{L} and if the remainder bound of all univariate functions in \mathcal{L} satisfies $r_g(A) = 0$ whenever the input model has degree 1 (this is satisfied for all operations in Table 2), then the first order interval superposition model $F_{f,X}$ computed by the above outlined arithmetic rules satisfies*

$$\max_{x \in X} \{ \text{diam}(F_{f,X}(x)) \} \leq \mathbf{O}\left(\frac{1}{N}\right)$$

for all bounded domains $X \subseteq \mathbb{I}^n$.

Given the above lemmata a formal proof of the above theorem is straightforward and skipped for the sake of brevity.

3 Higher Order Interval Superposition Arithmetic

This section generalizes the ideas from the previous section on first order interval superposition models to higher orders. Similar to the construction of multivariate orthogonal polynomials from univariate orthogonal polynomials, as for example used for the construction of Chebychev models [36], higher order extensions of interval superposition arithmetics can be constructed by using monomial combinations of the basis functions ϕ_i^j from (10). In the following, the notation

$$T_\alpha^\beta(x) = \prod_{i=1}^v \phi_{\alpha_i}^{\beta_i}(x) \quad (26)$$

is used to denote the corresponding new set of basis functions. Here, $v \in \mathbb{N}$, $v \leq n$, denotes the order of the basis functions. The symbols

$$\alpha = (\alpha_1, \dots, \alpha_v) \in S_n^v \quad \text{and} \quad \beta = (\beta_1, \dots, \beta_v) \in \{1, \dots, N\}^v$$

denote v -dimensional index tuples. Here, S_n^v denotes the set of sub-sequences of length v of the integer sequence $1, 2, \dots, n$. The size of this index set is given by

$$|S_n^v| = \binom{n}{v} = \frac{n!}{v!(n-v)!}.$$

In order to avoid confusion at this point, it is mentioned that α is *not* a multi-index as used in the context of higher order Taylor models, but merely an ordered sequence of integers from 1 to n picked without repetition. The associated interval superposition model of order v is given by

$$F_{f,X}(x) = \sum_{\alpha \in S_n^v} \sum_{\beta \in \{1, \dots, N\}^v} A_{\alpha}^{\beta} T_{\alpha}^{\beta}(x) \quad (27)$$

with interval valued coefficients $A_{\alpha}^{\beta} \in \mathbb{I}$. As the basis functions ϕ_i^j are piecewise constant functions, their monomial products T_{α}^{β} are piecewise constant functions, too. Thus, interval superposition models of order v are piecewise constant interval enclosure functions, which may, however, take different interval values on all of its N^n sub-branches of the domain X . The cost of storing an interval superposition model of order v is equal to

$$\binom{n}{v} N^v \times \text{the cost of storing a scalar interval in } \mathbb{I}.$$

Thus, the complexity of storing such objects is of order $\mathbf{O}\left(\frac{1}{v!} n^v N^v\right)$, which is for every given order v polynomial in n and N —in contrast to the associated exponentially large number N^n of sub-branches of the domain X . Similar to the notation for first order interval models, we introduce the notation

$$L_{\alpha}(A) = \min_{\beta \in \{1, \dots, N\}^v} \underline{A}_{\alpha}^{\beta} \quad \text{and} \quad U_{\alpha}(A) = \max_{\beta \in \{1, \dots, N\}^v} \overline{A}_{\alpha}^{\beta}, \quad \text{with} \quad A_{\alpha}^{\beta} = [\underline{A}_{\alpha}^{\beta}, \overline{A}_{\alpha}^{\beta}].$$

Simple range boundaries of $F_{f,X}$ are then given by

$$\lambda(A) = \sum_{\alpha \in S_n^v} L_{\alpha}(A) \quad \text{and} \quad \mu(A) = \sum_{\alpha \in S_n^v} U_{\alpha}(A),$$

which can be computed within polynomial run-time $\mathbf{O}\left(\frac{1}{v!} n^v N^v\right)$. An important difference to the first order case, however, is that these simple range boundaries are not tight in general as interval superposition models with order ≥ 2 are not separable. As a technical preparation for deriving composition rules for interval superposition arithmetics of order $v \geq 2$ the following section introduces the concept of dependency expansions.

3.1 Dependency Expansions

Let $g \in \mathcal{L}$ denote a univariate atom operation for which either a rational addition theorem or an inverse rational addition theorem is available. The goal of this section is to analyze expressions of the form

$$g\left(\omega + \sum_{i=1}^n x_i\right). \quad (28)$$

The argument of this expression is a superposition of n variables, $x \in \mathbb{R}^n$, and a constant offset $\omega \in \mathbb{R}$. If g admits a rational addition theorem, this theorem can be applied recursively in order to rewrite the above expression as a rational expression of $g(\omega), g(x_1), \dots, g(x_n)$.

However, this section asks the question how to expand the expression (28) into weighted sums of expressions of the form

$$g\left(\omega + \sum_{i=1}^k x_{\alpha_i}\right), \quad \alpha \in S_n^k,$$

where α can be any sub-sequence or order $k \leq v$ of the index sequence from $1, \dots, n$. In order to analyze this, the concept of *dependency expansions* is introduced.

Definition 2 For any given expansion order $v < n$, an equation of the form

$$g\left(\omega + \sum_{i=1}^n x_i\right) = \sum_{k=0}^v \sum_{\alpha \in S_n^k} (-1)^{v-k} \binom{n-k-1}{v-k} g\left(\omega + \sum_{i=1}^k x_{\alpha_i}\right) + \rho_g^v(x) \quad (29)$$

is called a dependency expansion of g . In analogy to multivariate Taylor expansions, the function $\rho_g^v(x)$ is called the remainder term.

The remainder term function $\rho_g^v(x)$ of a dependency expansion of order v has a remarkable property, namely, any of its mixed higher order derivatives are equal to zero as long as we differentiate with respect to at most v variables at the same time. The following lemma provides a formal proof of this powerful result.

Lemma 5 *If g is a smooth function in an open neighborhood of ω , then the remainder term ρ_g^v , $v \in \mathbb{N}$, $v < n$, of the dependency expansion of g (see Definition 2) is also smooth in this neighborhood and satisfies for all $\kappa \in S_n^v$ and all integers $m_1, m_2, \dots, m_v \in \mathbb{N}_0 = \{0, 1, 2, \dots\}$ the equation*

$$\frac{\partial^m}{\partial x_{\kappa_1}^{m_1} \partial x_{\kappa_2}^{m_2} \dots \partial x_{\kappa_v}^{m_v}} \rho_g^v(0) = 0 \quad \text{with} \quad m = \sum_{i=1}^v m_i.$$

Proof. Let $g^{(m)}$ denote the m -th derivative of the function g such that the associated mixed derivative of the left hand side of (29) can be written in the form

$$\frac{\partial^m}{\partial x_{\kappa_1}^{m_1} \partial x_{\kappa_2}^{m_2} \dots \partial x_{\kappa_v}^{m_v}} g\left(\omega + \sum_{i=1}^n x_i\right) \Big|_{x=0} = g^{(m)}(\omega). \quad (30)$$

Next, let $p \in S_n^\ell$ denote the sequence of indices that correspond to non-zero elements of the sequence m_1, \dots, m_v , i.e., such that $m_{p_i} \neq 0$ for all $i \in \{1, \dots, \ell\}$, where $\ell \leq v$ denotes the total number of non-zero elements of the integer sequence m . Moreover, let the sequence $\kappa' \in S_n^\ell$ be defined as $\kappa'_i = \kappa_{p_i}$ for all $i \in \{1, \dots, \ell\}$. This notation is useful for analyzing the derivative expression

$$\frac{\partial^m}{\partial x_{\kappa_1}^{m_1} \partial x_{\kappa_2}^{m_2} \dots \partial x_{\kappa_v}^{m_v}} g\left(\omega + \sum_{i=1}^k x_{\alpha_i}\right) \Big|_{x=0} = \begin{cases} g^{(m)}(\omega) & \text{if } \kappa' \text{ is a subsequence of } \alpha \\ 0 & \text{otherwise.} \end{cases} \quad (31)$$

If we have $k < \ell$, the sequence κ' has more elements than the sequence α and, consequently, κ' cannot possibly be a subsequence of α . Next, for the case $k \geq \ell$ there are exactly

$$\binom{n-\ell}{k-\ell} = \frac{(n-\ell)!}{(k-\ell)!(n-k)!}$$

sequences $\alpha \in S_n^k$ that contain the sequence $\kappa' \in S_n^\ell$ as a sub-sequence. This follows from the fact that we can construct such sequences $\alpha \in S_n^k$ by starting with the ℓ elements of κ' and choosing the remaining $k - \ell$ elements from the set $\{1, \dots, n\} \setminus \kappa'$, which contains $n - \ell$ elements. Because there are “ $(n - \ell)$ -choose- $(k - \ell)$ ” possibilities for choosing the remaining elements, the above statement follows. Being at this point, it follows from (31) that

$$\begin{aligned} & \sum_{\alpha \in S_n^k} (-1)^{v-k} \binom{n-k-1}{v-k} \frac{\partial^m}{\partial x_{\kappa_1}^{m_1} \partial x_{\kappa_2}^{m_2} \dots \partial x_{\kappa_v}^{m_v}} g \left(\omega + \sum_{i=1}^k x_{\alpha_i} \right) \Big|_{x=0} \\ &= \begin{cases} (-1)^{v-k} \binom{n-k-1}{v-k} \binom{n-\ell}{k-\ell} g^{(m)}(\omega) & \text{if } k \geq \ell \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \quad (32)$$

A combination of (29), (30), and (32) yields the following formula for the derivative of the remainder term,

$$\frac{\partial^m}{\partial x_{\kappa_1}^{m_1} \partial x_{\kappa_2}^{m_2} \dots \partial x_{\kappa_v}^{m_v}} \rho_g^v(x) \Big|_{x=0} = \underbrace{\left[1 - \sum_{k=\ell}^v (-1)^{v-k} \binom{n-k-1}{v-k} \binom{n-\ell}{k-\ell} \right]}_{=N(n,v,\ell)} g^{(m)}(\omega). \quad (33)$$

It remains to show that the expression in the bracket on the right hand of this equation, denoted by $N(n, v, \ell)$, is equal to zero for all integers n, v, ℓ with $\ell \leq v < n$. The main idea is to use a backward induction over ℓ . Here, the induction start is obtained by observing that

$$N(n, v, v) = 1 - \binom{n-k-1}{0} \binom{n-v}{0} = 1 - 1 * 1 = 0.$$

The induction assumption is that the equation $N(n, v, \ell) = 0$ holds for a given $\ell > 0$. The main idea for working out the induction step is to use Pascal's recursion formula for binomial coefficients in order to establish the equation

$$\begin{aligned} N(n, v, \ell-1) &= 1 - \sum_{k=\ell-1}^v (-1)^{v-k} \binom{n-k-1}{v-k} \binom{n-\ell+1}{k-\ell+1} \\ &= 1 - (-1)^{v-\ell+1} \binom{n-\ell}{v-\ell+1} \\ &\quad - \sum_{k=\ell}^v (-1)^{v-k} \binom{n-k-1}{v-k} \left[\binom{n-\ell}{k-\ell+1} + \binom{n-\ell}{k-\ell} \right] \\ &= - \sum_{k=\ell-1}^v (-1)^{v-k} \binom{n-k-1}{v-k} \binom{n-\ell}{k-\ell+1}. \end{aligned} \quad (34)$$

In the latter equation the induction assumption has been substituted. In order to transform the remaining expression further, the product of the binomial coefficients can be re-arranged

as

$$\begin{aligned} N(n, v, \ell - 1) &= - \sum_{k=\ell-1}^v (-1)^{v-k} \binom{n-k-1}{v-k} \binom{n-\ell}{k-\ell+1} \\ &= - \sum_{k=\ell-1}^v (-1)^{v-k} \binom{v-\ell+1}{v-k} \binom{n-\ell}{v-\ell+1}. \end{aligned} \quad (35)$$

The correctness of this equation can be checked easily by expanding the definition of the binomial coefficients and collecting terms on both sides. The reason why the latter transformation is useful is that it removes the dependence on the summation index k from one of the binomial factors. Thus, we can move this factor out of the sum finding

$$N(n, v, \ell - 1) = - \binom{n-\ell}{v-\ell+1} \underbrace{\left[\sum_{k=\ell-1}^v (-1)^{v-k} \binom{v-\ell+1}{v-k} \right]}_{=0}. \quad (36)$$

Here, the sum in the bracket is an alternating sum over the integers in the $(v-\ell+1)$ -th row of Pascal's triangle, which equals out to 0. Thus, we find $N(n, v, \ell - 1) = 0$, which concludes the induction proof and, together with (33), the proof of the lemma. \square

In addition to the technical result of the above lemma, it is useful to observe that the sum in (29) can be resorted with respect to its dependencies. In the following, we use the notation

$$\mathcal{S}_\alpha^k = \{ \alpha' \in S_n^k \mid \forall i \in \{1, \dots, k\}, \exists j \in \{1, \dots, v\} : \alpha'_i = \alpha_j \}$$

to denote the set of sub-sequences of a given $\alpha \in S_n^v$ with length $k \leq v$. Notice that $\mathcal{S}_\alpha^v = S_n^v$.

Lemma 6 *Let $g \in \mathcal{L}$ be any given univariate atom function. The following re-arrangement equation holds for all $x \in \mathbb{R}^n$, all $\omega \in \mathbb{R}$, all $v \in \mathbb{N}$ with $v < n$, and all $n \in \mathbb{N}$:*

$$\begin{aligned} &\sum_{k=0}^v \sum_{\alpha \in S_n^k} (-1)^{v-k} \binom{n-k-1}{v-k} g \left(\omega + \sum_{i=1}^k x_{\alpha_i} \right) \\ &= \sum_{\alpha \in S_n^v} \left\{ \sum_{k=0}^v (-1)^{v-k} \frac{n-v}{n-k} \sum_{\alpha' \in \mathcal{S}_\alpha^k} g \left(\omega + \sum_{i=1}^k x_{\alpha'_i} \right) \right\}. \end{aligned} \quad (37)$$

Proof. A proof of the lemma can be obtained by counting the summands in (37) in two different ways. On the one hand, for a given sequence $\alpha' \in S_n^k$, there are

$$\binom{n-k}{v-k}$$

sequences $\alpha \in S_n^v$, which contain α' as a subsequence. This is due to the fact that there are $v-k$ components of α , which can be picked from the $n-k$ integers from 1 to n that are not among the components of α' . And, on the other hand, the recursion formula

$$\binom{n-k-1}{v-k} = \frac{n-v}{n-k} \binom{n-k}{v-k}$$

holds. The proof of the lemma follows by combining these two statements. \square

3.2 Composition Rules

This section presents a generic framework for deriving composition rules for higher order interval superposition models. Similar to the first order case,

$$F_{h,X}(x) = \sum_{\alpha \in S_n^v} \sum_{\beta \in \{1, \dots, N\}^v} A_{\alpha}^{\beta} T_{\alpha}^{\beta}(x)$$

denotes the interval superposition model of the function $h : X \rightarrow \mathbb{R}$ while $g \in \mathcal{L}$ denotes a univariate atom function. The goal is to compute coefficients $C_{\alpha}^{\beta} \in \mathbb{I}$ of an interval superposition model

$$F_{f,X}(x) = \sum_{\alpha \in S_n^v} \sum_{\beta \in \{1, \dots, N\}^v} C_{\alpha}^{\beta} T_{\alpha}^{\beta}(x) \quad \text{enclosing the function } f = g \circ h$$

on the domain X . Algorithm 3 proposes a generic composition rule for interval superposition models of order $v \geq 2$. Notice that this algorithm is based on the following notation conventions.

1. For any sequence $\gamma \in S_m^k$ the notation $|\gamma| = k \leq m$ is used to denote its length.
2. For any sequence $j = (j_1, j_2, \dots, j_m)$ and any given sequence $\alpha \in S_m^k$, $k \leq m$, the notation

$$j_{\alpha} = (j_{\alpha_1}, j_{\alpha_2}, \dots, j_{\alpha_k}) \in \{1, \dots, N\}^k$$

is used to denote the sub-sequence that is obtained by picking all components of the index vector j , whose index is an element of the sequence α .

3. For two sequences α, α' we write $\alpha' \subset \alpha$ if α' is a subsequence of α but $\alpha \neq \alpha'$. Moreover, we write $\alpha' \subseteq \alpha$ if either $\alpha' \subset \alpha$ or $\alpha' = \alpha$.
4. We use the notation

$$\mathcal{S}_{\alpha} = \{\alpha' \mid \alpha' \subset \alpha\} = \bigcup_{k \in \{1, \dots, v-1\}} \mathcal{S}_{\alpha}^k$$

to denotes the set of all sub-sequence of a given sequence $\alpha \in S_n^v$. Additionally, we define

$$\overline{\mathcal{S}}_{\alpha} = \{\alpha' \mid \alpha' \subseteq \alpha\} = \mathcal{S}_{\alpha} \cup \{\alpha\} = \bigcup_{k \in \{1, \dots, v\}} \mathcal{S}_{\alpha}^k.$$

5. For any given sequence $\alpha \in S_m^k$ and any sub-sequence $\gamma \in \mathcal{S}_{\alpha}$ we associate a quotient sequence, denoted by $\alpha/\gamma \in S_{|\alpha|}^{|\gamma|}$, which is defined uniquely by the relation

$$\alpha_{\alpha/\gamma} = \gamma.$$

The following theorem establishes the fact that Algorithm 3 yields an enclosure of the univariate composition $f = g \circ h$. Here, the main idea is to use a dependency expansion with respect to the dependencies of the inner superposition model and to apply Lemma 6 in order to show that the remainder bound is valid.

Algorithm 3: Composition rule of higher order interval superposition arithmetic

Input:

Order $v \geq 2$, coefficients $A_\alpha^\beta = [\underline{A}_\alpha^\beta, \overline{A}_\alpha^\beta] \in \mathbb{I}$, input model $F_{h,X}^f$, and a univariate function $g \in \mathcal{L}$.

Main Steps:

1. Choose for all $\alpha \in S_n^v$ suitable central points $a_\alpha \in \mathbb{R}$ such that

$$L_\alpha(A) \leq a_\alpha \leq U_\alpha(A) \quad \text{and set} \quad \omega = \sum_{\alpha \in S_n^v} a_\alpha.$$

2. Choose higher order variation points $a_{\alpha,\gamma}^\beta \in \mathbb{R}$, $\gamma \in \mathcal{S}_\alpha$, $\beta \in \{1, \dots, N\}^{|\gamma|}$, and compute

$$L_{\alpha,\gamma} = \min_{\beta \in \{1, \dots, N\}^{|\gamma|}} a_{\alpha,\gamma}^\beta \quad \text{and} \quad U_{\alpha,\gamma} = \max_{\beta \in \{1, \dots, N\}^{|\gamma|}} a_{\alpha,\gamma}^\beta \quad \text{and set} \quad \omega_\gamma^\beta = \sum_{\substack{\alpha \in S_n^v \\ \alpha \supset \gamma}} a_{\alpha,\gamma}^\beta.$$

Additionally, compute for all $\alpha \in S_n^v$ the inner remainder bounds

$$L_{\alpha,\alpha} = \min_{\beta \in \{1, \dots, N\}^v} \underline{A}_\alpha^\beta, \quad U_{\alpha,\alpha} = \max_{\beta \in \{1, \dots, N\}^v} \overline{A}_\alpha^\beta \quad \text{with} \quad \Delta_\alpha^\beta = A_\alpha^\beta - a_\alpha - \sum_{\gamma \in \mathcal{S}_\alpha} a_{\alpha,\gamma}^{\beta/\gamma}.$$

3. Choose a remainder bound $r_g(A) \geq 0$ such that for all $\delta_{\alpha,\gamma}$ with $\delta_{\alpha,\gamma} \in [L_{\alpha,\gamma}, U_{\alpha,\gamma}]$, $\gamma \subseteq \alpha$,

$$r_g(A) \geq \left| g \left(\omega + \sum_{\alpha \in S_n^v} \sum_{\gamma \in \mathcal{S}_\alpha} \delta_{\alpha,\gamma} \right) - \sum_{k=0}^v \sum_{\alpha' \in S_n^k} (-1)^{v-k} \binom{n-k-1}{v-k} g \left(\omega + \sum_{\gamma \in \mathcal{S}_{\alpha'}} \sum_{\substack{\alpha \in S_n^v \\ \alpha \supseteq \gamma}} \delta_{\alpha,\gamma} \right) \right|$$

4. Compute for all $\alpha \in S_n^v$ and all $\beta \in \{1, \dots, N\}^v$ the interval valued coefficients

$$C_\alpha^\beta = g \left(\omega + \sum_{\gamma \in \mathcal{S}_\alpha} \omega_\gamma^{\beta/\gamma} + \Delta_\alpha^\beta \right) + \sum_{k=0}^{v-1} (-1)^{v-k} \frac{n-v}{n-k} \sum_{\alpha' \in \mathcal{S}_\alpha^k} g \left(\omega + \sum_{\gamma \in \mathcal{S}_{\alpha'}} \omega_\gamma^{\beta/\gamma} \right).$$

Pick a suitable $\hat{\alpha} \in S_n^v$ and add the remainder, $C_{\hat{\alpha}}^\beta \leftarrow C_{\hat{\alpha}}^\beta + r_g(A)$ for all $\beta \in \{1, \dots, N\}^v$.

Output: The coefficients C_α^β of a v -th order interval superposition model $F_{f,X}$ of the function $f = g \circ h$.

Theorem 5 If $F_{h,X}(x) = \sum_{\alpha \in S_n^v} \sum_{\beta \in \{1, \dots, N\}^v} A_\alpha^\beta T_\alpha^\beta(x)$ is an enclosure function of h on X and if the coefficient C_α^β are computed by Algorithm 3, then the function

$$F_{f,X}(x) = \sum_{\alpha \in S_n^v} \sum_{\beta \in \{1, \dots, N\}^v} C_\alpha^\beta T_\alpha^\beta(x)$$

is an enclosure function of $f = g \circ h$.

Proof. For every $x \in X$ there exist indices $j_1, j_2, \dots, j_n \in \{1, \dots, N\}$ such that $x_i \in X_i^{j_i}$. As $F_{h,X}$ is an enclosure of the function h , there exist points $y_\alpha \in A_\alpha^{j_\alpha}$ such that

$$h(x) = \sum_{\alpha \in S_n^v} y_\alpha.$$

Here, we recall that $j_\alpha = (j_{\alpha_1}, \dots, j_{\alpha_v})$ denotes the sub-sequence of the index sequence j that is associated with $\alpha \in S_n^v$. Next, the image point $f(x)$ can be represented as

$$f(x) = g(h(x)) = g\left(\sum_{\alpha \in S_n^v} y_\alpha\right) = g\left(\omega + \sum_{\alpha \in S_n^v} \{y_\alpha - a_\alpha\}\right), \quad (38)$$

where we have used that the sum of the central points is given by $\omega = \sum_{\alpha \in S_n^v} a_\alpha$. Similarly, we add and subtract the sum of the higher order variation points in order to establish the equation

$$f(x) = g\left(\omega + \sum_{\alpha \in S_n^v} \sum_{\gamma \in \mathcal{J}_\alpha} a_{\alpha,\gamma}^{(j_\alpha)\alpha/\gamma} + \sum_{\alpha \in S_n^v} \left\{y_\alpha - a_\alpha - \sum_{\gamma \in \mathcal{J}_\alpha} a_{\alpha,\gamma}^{(j_\alpha)\alpha/\gamma}\right\}\right). \quad (39)$$

In the next step, we apply a dependency expansion to the right hand of the above equation. This yields

$$\begin{aligned} f(x) &= g\left(\omega + \sum_{\alpha \in S_n^v} \sum_{\gamma \in \mathcal{J}_\alpha} a_{\alpha,\gamma}^{(j_\alpha)\alpha/\gamma} + \sum_{\alpha \in S_n^v} \left\{y_\alpha - a_\alpha - \sum_{\gamma \in \mathcal{J}_\alpha} a_{\alpha,\gamma}^{(j_\alpha)\alpha/\gamma}\right\}\right) \\ &= \sum_{\alpha \in S_n^v} \left\{g\left(\omega + \sum_{\gamma \in \mathcal{J}_\alpha} \sum_{\alpha' \supset \gamma} a_{\alpha',\gamma}^{(j_\alpha)\alpha/\gamma} + \left(y_\alpha - a_\alpha - \sum_{\gamma \in \mathcal{J}_\alpha} a_{\alpha,\gamma}^{(j_\alpha)\alpha/\gamma}\right)\right)\right. \\ &\quad \left.+ \sum_{k=0}^{v-1} (-1)^{v-k} \frac{n-v}{n-k} \sum_{\alpha' \in \mathcal{J}_\alpha^k} g\left(\omega + \sum_{\gamma \in \mathcal{J}_{\alpha'}} \sum_{\substack{\alpha'' \in S_n^v \\ \alpha'' \supset \gamma}} a_{\alpha'',\gamma}^{(j_\alpha)\alpha/\gamma}\right)\right\} - \rho_g^v, \quad (40) \end{aligned}$$

where the remainder term ρ_g^v is simply defined in such a way that the above equation holds; that is

$$\begin{aligned} \rho_g^v(x) &= \sum_{\alpha \in S_n^v} \left\{g\left(\omega + \sum_{\gamma \in \mathcal{J}_\alpha} \sum_{\substack{\alpha' \in S_n^v \\ \alpha' \supset \gamma}} a_{\alpha',\gamma}^{(j_\alpha)\alpha/\gamma} + \left(y_\alpha - a_\alpha - \sum_{\gamma \in \mathcal{J}_\alpha} a_{\alpha,\gamma}^{(j_\alpha)\alpha/\gamma}\right)\right)\right. \\ &\quad \left.+ \sum_{k=0}^{v-1} (-1)^{v-k} \frac{n-v}{n-k} \sum_{\alpha' \in \mathcal{J}_\alpha^k} g\left(\omega + \sum_{\gamma \in \mathcal{J}_{\alpha'}} \sum_{\substack{\alpha'' \in S_n^v \\ \alpha'' \supset \gamma}} a_{\alpha'',\gamma}^{(j_\alpha)\alpha/\gamma}\right)\right\} \\ &\quad - g\left(\omega + \sum_{\alpha \in S_n^v} \sum_{\gamma \in \mathcal{J}_\alpha} a_{\alpha,\gamma}^{(j_\alpha)\alpha/\gamma} + \sum_{\alpha \in S_n^v} \left\{y_\alpha - a_\alpha - \sum_{\gamma \in \mathcal{J}_\alpha} a_{\alpha,\gamma}^{(j_\alpha)\alpha/\gamma}\right\}\right). \quad (41) \end{aligned}$$

In order to simplify this expression further we introduce the shorthands

$$\delta_{\alpha,\gamma} = a_{\alpha,\gamma}^{(j_\alpha)\alpha/\gamma} \in [L_{\alpha,\gamma}, U_{\alpha,\gamma}]$$

for all $\gamma \in \mathcal{J}_\alpha$ as well as

$$\delta_{\alpha,\alpha} = y_\alpha - a_\alpha - \sum_{\gamma \in \mathcal{J}_\alpha} a_{\alpha,\gamma}^{(j_\alpha)\alpha/\gamma} \in [L_{\alpha,\alpha}, U_{\alpha,\alpha}].$$

Now, we can use this notation and apply Lemma 6, which yields the equation

$$\begin{aligned} \rho_g^v(x) &= \sum_{k=0}^v \sum_{\alpha' \in S_n^k} (-1)^{v-k} \binom{n-k-1}{v-k} g \left(\omega + \sum_{\gamma \in \mathcal{J}_{\alpha'}} \sum_{\substack{\alpha \in S_n^v \\ \alpha \supseteq \gamma}} \delta_{\alpha, \gamma} \right) \\ &\quad - g \left(\omega + \sum_{\alpha \in S_n^v} \sum_{\gamma \in \mathcal{J}_{\alpha}} \delta_{\alpha, \gamma} \right). \end{aligned} \quad (42)$$

In this form it becomes clear that we have $\rho_g^v(x) \leq r_g(A)$, as the remainder bound in Step 3 of Algorithm 3 is constructed in such a way that this inequality holds. Moreover, (40) can be simplified further by substituting the definition of the summation variables ω_{γ}^{β} as well as by using that

$$y_{\alpha} - a_{\alpha} - \sum_{\gamma \in \mathcal{J}_{\alpha}} a_{\alpha, \gamma}^{(j_{\alpha})\alpha/\gamma} \in \Delta_{\alpha}^{j_{\alpha}}.$$

This yields

$$\begin{aligned} f(x) &\in \sum_{\alpha \in S_n^v} \left\{ g \left(\omega + \sum_{\gamma \in \mathcal{J}_{\alpha}} \omega_{\gamma}^{(j_{\alpha})\alpha/\gamma} + \Delta_{\alpha}^{j_{\alpha}} \right) \right. \\ &\quad \left. + \sum_{k=0}^{v-1} (-1)^{v-k} \frac{n-v}{n-k} \sum_{\alpha' \in \mathcal{J}_{\alpha}^k} g \left(\omega + \sum_{\gamma \in \mathcal{J}_{\alpha'}} \omega_{\gamma}^{(j_{\alpha})\alpha/\gamma} \right) \right\} + \rho_g^v(x) \\ &\subseteq \sum_{\alpha \in S_n^v} C_{\alpha}^{j_{\alpha}} = \sum_{\alpha \in S_n^v} \sum_{\beta \in \{1, \dots, N\}^v} C_{\alpha}^{\beta} T_{\alpha}^{\beta}(x) = F_{f, X}(x). \end{aligned} \quad (43)$$

As this construction is valid for any $x \in X$, this implies the statement of the theorem. \square

3.3 Extension of the composition rule for bivariate operation

The computation of the sum of two interval superposition models

$$F_{h, X}(x) = \sum_{\alpha \in S_n^v} \sum_{\beta \in \{1, \dots, N\}^v} A_{\alpha}^{\beta} T_{\alpha}^{\beta}(x) \quad \text{and} \quad F_{g, X}(x) = \sum_{\alpha \in S_n^v} \sum_{\beta \in \{1, \dots, N\}^v} B_{\alpha}^{\beta} T_{\alpha}^{\beta}(x)$$

can be implemented trivially by adding all coefficients, i.e., the coefficients of the sum are given by

$$C_{\alpha}^{\beta} = A_{\alpha}^{\beta} + B_{\alpha}^{\beta}.$$

Moreover, the product rule of higher order interval superposition arithmetic is summarized in Algorithm 4.

Algorithm 4: Product rule of higher order interval superposition arithmetic**Input:**

Order $v \geq 2$, coefficients $A_\alpha^\beta \in \mathbb{I}^2$ of the factors.

Main Steps:

1. Choose for all $\alpha \in S_n^v$ suitable central points $a_\alpha \in \mathbb{R}^2$ such that

$$L_\alpha(A) \leq a_\alpha \leq U_\alpha(A) \quad \text{and set} \quad \omega = \sum_{\alpha \in S_n^v} a_\alpha.$$

2. Choose higher order variation points $a_{\alpha,\gamma}^\beta \in \mathbb{R}^2$, $\gamma \in \mathcal{S}_\alpha$, $\beta \in \{1, \dots, N\}^{|\gamma|}$, and compute

$$\rho_{\alpha,\gamma} = \begin{pmatrix} \max_{\beta \in \{1, \dots, N\}^{|\gamma|}} [a_{\alpha,\gamma}^\beta]_1 - \min_{\beta \in \{1, \dots, N\}^{|\gamma|}} [a_{\alpha,\gamma}^\beta]_1 \\ \max_{\beta \in \{1, \dots, N\}^{|\gamma|}} [a_{\alpha,\gamma}^\beta]_2 - \min_{\beta \in \{1, \dots, N\}^{|\gamma|}} [a_{\alpha,\gamma}^\beta]_2 \end{pmatrix} \quad \text{set} \quad \omega_\gamma^\beta = \sum_{\substack{\alpha \in S_n^v \\ \alpha \supset \gamma}} a_{\alpha,\gamma}^\beta$$

and

$$\rho_{\alpha,\alpha} = \begin{pmatrix} \max_{\beta \in \{1, \dots, N\}^v} \max_{\delta \in [\Delta_\alpha^\beta]_1} |\delta| \\ \max_{\beta \in \{1, \dots, N\}^v} \max_{\delta \in [\Delta_\alpha^\beta]_2} |\delta| \end{pmatrix} \quad \text{with} \quad \Delta_\alpha^\beta = A_\alpha^\beta - a_\alpha - \sum_{\gamma \in \mathcal{S}_\alpha} a_{\alpha,\gamma}^{\beta_{\alpha/\gamma}}.$$

3. Compute the remainder bound

$$r_g(A) = \left[\sum_{\alpha \in S_n^v} \sum_{\gamma \in \mathcal{S}_\alpha} \rho_{\alpha,\gamma} \right]_1 \left[\sum_{\alpha \in S_n^v} \sum_{\gamma \in \mathcal{S}_\alpha} \rho_{\alpha,\gamma} \right]_2 \\ - \sum_{k=1}^v \sum_{\alpha' \in S_n^k} (-1)^{v-k} \binom{n-k-1}{v-k} \left[\sum_{\gamma \in \mathcal{S}_{\alpha'}} \sum_{\substack{\alpha \in S_n^v \\ \alpha \supset \gamma}} \rho_{\alpha,\gamma} \right]_1 \left[\sum_{\gamma \in \mathcal{S}_{\alpha'}} \sum_{\substack{\alpha \in S_n^v \\ \alpha \supset \gamma}} \rho_{\alpha,\gamma} \right]_2$$

4. Compute for all $\alpha \in S_n^v$ and all $\beta \in \{1, \dots, N\}^v$ the interval valued coefficients

$$C_\alpha^\beta = \left[\omega + \sum_{\gamma \in \mathcal{S}_\alpha} \omega_\gamma^{\beta_{\alpha/\gamma}} + \Delta_\alpha^\beta \right]_1 \left[\omega + \sum_{\gamma \in \mathcal{S}_\alpha} \omega_\gamma^{\beta_{\alpha/\gamma}} + \Delta_\alpha^\beta \right]_2 \quad (44) \\ + \sum_{k=0}^{v-1} (-1)^{v-k} \frac{n-v}{n-k} \sum_{\alpha' \in \mathcal{S}_\alpha^k} \left[\omega + \sum_{\gamma \in \mathcal{S}_{\alpha'}} \omega_\gamma^{\beta_{\alpha/\gamma}} \right]_1 \left[\omega + \sum_{\gamma \in \mathcal{S}_{\alpha'}} \omega_\gamma^{\beta_{\alpha/\gamma}} \right]_2.$$

Pick a suitable $\hat{\alpha} \in S_n^v$ and add the remainder, $C_{\hat{\alpha}}^\beta \leftarrow C_{\hat{\alpha}}^\beta + r_g(A)$ for all $\beta \in \{1, \dots, N\}^v$.

Output: The coefficients C_α^β of an interval superposition model $F_{f,X}$ of the product of the input models.

Theorem 6 *If the v -th order interval superposition models*

$$F_{h,X}(x) = \sum_{\alpha \in S_n^v} \sum_{\beta \in \{1, \dots, N\}^v} A_\alpha^\beta T_\alpha^\beta(x) \quad \text{and} \quad F_{g,X}(x) = \sum_{\alpha \in S_n^v} \sum_{\beta \in \{1, \dots, N\}^v} B_\alpha^\beta T_\alpha^\beta(x)$$

are enclosure functions of the functions h and g on X and if the coefficient C_α^β are computed by Algorithm 4, then the function

$$F_{f,X}(x) = \sum_{\alpha \in S_n^v} \sum_{\beta \in \{1, \dots, N\}^v} C_\alpha^\beta T_\alpha^\beta(x)$$

is an enclosure of the product of g and h , $f(x) = g(x)h(x)$.

Proof. There are two notable differences between Algorithm 3 and Algorithm 4. Firstly, Algorithm 4 uses vector valued central and higher order variation points, as the product operation has two input arguments. And secondly, Algorithm 4 works out an explicit remainder bound in Step 3. Thus, the statement of this theorem can be obtained by using exactly the same argumentation and steps as the proof of Theorem 5 with the only difference that the central and higher order variation points are now vectors of dimension 2. Here, the remainder bound

$$\begin{aligned} r_g(A) = & \left[\sum_{\alpha \in S_n^v} \sum_{\gamma \in \mathcal{T}_\alpha} \rho_{\alpha,\gamma} \right]_1 \left[\sum_{\alpha \in S_n^v} \sum_{\gamma \in \mathcal{T}_\alpha} \rho_{\alpha,\gamma} \right]_2 \\ & - \sum_{k=0}^v \sum_{\alpha' \in S_n^k} (-1)^{v-k} \binom{n-k-1}{v-k} \left[\sum_{\gamma \in \mathcal{T}_{\alpha'}} \sum_{\substack{\alpha \in S_n^v \\ \alpha \supseteq \gamma}} \rho_{\alpha,\gamma} \right]_1 \left[\sum_{\gamma \in \mathcal{T}_{\alpha'}} \sum_{\substack{\alpha \in S_n^v \\ \alpha \supseteq \gamma}} \rho_{\alpha,\gamma} \right]_2 \end{aligned}$$

from Step 3 of Algorithm 4 satisfies the inequality

$$\begin{aligned} r_g(A) \geq & \left[\omega + \sum_{\alpha \in S_n^v} \sum_{\gamma \in \mathcal{T}_\alpha} \delta_{\alpha,\gamma} \right]_1 \left[\omega + \sum_{\alpha \in S_n^v} \sum_{\gamma \in \mathcal{T}_\alpha} \delta_{\alpha,\gamma} \right]_2 \\ & - \sum_{k=0}^v \sum_{\alpha' \in S_n^k} (-1)^{v-k} \binom{n-k-1}{v-k} \left[\omega + \sum_{\gamma \in \mathcal{T}_{\alpha'}} \sum_{\substack{\alpha \in S_n^v \\ \alpha \supseteq \gamma}} \delta_{\alpha,\gamma} \right]_1 \left[\omega + \sum_{\gamma \in \mathcal{T}_{\alpha'}} \sum_{\substack{\alpha \in S_n^v \\ \alpha \supseteq \gamma}} \delta_{\alpha,\gamma} \right]_2 \end{aligned}$$

for all $\delta_{\alpha,\gamma} \in \mathbb{R}^2$ with $|\delta_{\alpha,\gamma}|_1 \leq [\rho_{\alpha,\gamma}]_1$ and $|\delta_{\alpha,\gamma}|_2 \leq [\rho_{\alpha,\gamma}]_2$, which is obtained by extending Algorithm 3 for vector valued arguments substituting $g(x) = x_1 x_2$. The latter inequality follows by expanding the right-hand of the above inequality (thereby all monomials containing ω cancel out), bounding all monomials in $\delta_{\alpha,\gamma}$ separately by using standard interval arithmetic, and re-bracketing the result. By using this remainder bound inequality the statement of theorem follows, as all other arguments are completely analogous to the proof of Theorem 5. \square

3.4 Construction of central and higher order variation points

The optimal choice of the central points a_α and higher order variation points $a_{\alpha,\gamma}^\beta$ in Algorithm 3 depends on the particular atom operation g . However, as the derivation of tailored higher order variation points for particular g can be rather cumbersome, this section proposes a generic and computationally tractable (but in general sub-optimal) way of computing central points and higher order variation points without exploiting properties of g .

Algorithm 5: Recursive averaging algorithm for computing central and higher order variation points**Input:**

Order $v \geq 2$, coefficients $A_\alpha^\beta = [\underline{A}_\alpha^\beta, \overline{A}_\alpha^\beta] \in \mathbb{I}$.

Initialization:

- Set $k = 1$. Compute the mid points $c_\alpha^\beta = (\overline{A}_\alpha^\beta - \underline{A}_\alpha^\beta) / 2$.
- Compute the central points by averaging

$$a_\alpha = \frac{1}{N^v} \sum_{\beta \in \{1, \dots, N\}^v} c_\alpha^\beta.$$

Repeat:

1. Compute for all $\gamma \in \mathcal{S}_\alpha^k$ and all $\beta \in \{1, \dots, N\}^k$ the associated running average

$$a_{\alpha, \gamma}^\beta = \frac{1}{N^{v-k}} \sum_{\substack{\beta' \in \{1, \dots, N\}^v \\ \beta'_{\alpha/\gamma} = \beta}} \left(c_{\alpha}^{\beta'} - a_\alpha - \sum_{\substack{\gamma' \in \mathcal{S}_\gamma \\ |\gamma'| \leq k-1}} a_{\alpha, \gamma'}^{\beta_{\gamma/\gamma'}} \right).$$

2. If $k = v - 1$, stop. Otherwise, set $k \leftarrow k + 1$ and continue with Step 1.

Output: Central point a_α and higher order variation points $a_{\alpha, \gamma}^\beta$.

The corresponding method is based on a recursive averaging strategy that is outlined in Algorithm 5. The advantage of Algorithm 5 is that its computational complexity is of order $\mathcal{O}(\frac{1}{v!} n^v N^v)$. In other words, Algorithm 5 is a polynomial run-time algorithm. Moreover, the central point a_α and higher order variation points $a_{\alpha, \gamma}^\beta$ that are computed by Algorithm 5 are the minimizer of the least-squares sum

$$\sum_{\beta \in \{1, \dots, N\}^v} \left[\text{diam} \left(A_\alpha^\beta - a_\alpha - \sum_{\gamma \in \mathcal{S}_\alpha} a_{\alpha, \gamma}^{\beta_{\alpha/\gamma}} \right) \right]^2.$$

This statement can be established easily by substituting the average expression for the central and higher order variation points from Algorithm 5 into the first optimality condition of the above least-squares optimization problem. Consequently, Algorithm 5 minimizes the least-squares average diameter of the remainder intervals Δ_α^β that are computed in Step 2 of Algorithm 3. An alternative way of choosing the central and higher order variation points could be obtained by choosing the minimizers of the maximum remainder interval diameter,

$$\max_{\beta \in \{1, \dots, N\}^v} \text{diam} \left(A_\alpha^\beta - a_\alpha - \sum_{\gamma \in \mathcal{S}_\alpha} a_{\alpha, \gamma}^{\beta_{\alpha/\gamma}} \right).$$

However, although this is still a convex optimization problem, which could—at least in principle—be solved approximately in polynomial run-time, this optimization problem has the disadvantage that no explicit recursion rule is available to compute its minimizers. Therefore, we will in this paper implement the simple and computationally efficient recursion rule from Algorithm 5, although minimizing the maximum diameter might in some cases lead

to tighter remainder bounds. For the implementation of the product rule from Algorithm 4 the above recursive averaging algorithm can be used to generate the central and higher order variation points for both input models, i.e., Algorithm 5 is run once for the input coefficients A_α^β of the first factor and then once more replacing A_α^β with B_α^β in order to compute the central and higher order variation points of the second factor.

3.5 Construction of remainder bounds

The basic strategy for the derivation of remainder bounds $r_g(A)$ for Algorithm 3 is analogous to the derivation of remainder bounds for the composition rule of the first order interval superposition arithmetic. The main idea is to bound the term

$$\begin{aligned} \hat{\rho}_g(\delta) = & g \left(\omega + \sum_{\alpha \in S_n^v} \sum_{\gamma \in \overline{\mathcal{T}}_\alpha} \delta_{\alpha,\gamma} \right) \\ & - \sum_{k=0}^v \sum_{\alpha' \in S_n^k} (-1)^{v-k} \binom{n-k-1}{v-k} g \left(\omega + \sum_{\gamma \in \overline{\mathcal{T}}_{\alpha'}} \sum_{\substack{\alpha \in S_n^v \\ \alpha \supseteq \gamma}} \delta_{\alpha,\gamma} \right) \end{aligned} \quad (45)$$

on the given interval domain, $\delta_{\alpha,\gamma} \in [L_{\alpha,\gamma}, U_{\alpha,\gamma}]$, by using addition theorems, symbolic simplification, as well as standard interval arithmetic. In order to impose accuracy requirements on the conservatism of the bound $r_g(A) \geq \hat{\rho}_g(\delta)$, the following local and global properties of the function $\hat{\rho}_g$ are established.

Lemma 7 *Let $g \in \mathcal{L}$ be a (locally) smooth function. If the bounds $L_{\alpha,\gamma}$ and $U_{\alpha,\gamma}$ satisfy the natural local convergence condition*

$$U_{\alpha,\gamma} - L_{\alpha,\gamma} = \mathbf{O} \left(\text{diam}(X)^{|\gamma|} \right)$$

for all $\alpha \in S_n^v$ and all $\gamma \in \overline{\mathcal{T}}_\alpha$, then the dependency expansion error function $\hat{\rho}_g$ satisfies

$$\max_{\delta, \delta_{\alpha,\gamma} \in [L_{\alpha,\gamma}, U_{\alpha,\gamma}]} \hat{\rho}_g(\delta) = \mathbf{O} \left(\text{diam}(X)^{v+1} \right).$$

The proof of the above lemma can be found below.

Lemma 8 *If there exists a sequence $\alpha' \in S_n^v$ such that $L_{\alpha,\gamma} = U_{\alpha,\gamma} = 0$ for all $\alpha \in S_n^v$ and all $\gamma \in \overline{\mathcal{T}}_\alpha$ with $\gamma \notin \overline{\mathcal{T}}_{\alpha'}$, then the dependency expansion error vanishes,*

$$\max_{\delta, \delta_{\alpha,\gamma} \in [L_{\alpha,\gamma}, U_{\alpha,\gamma}]} \hat{\rho}_g(\delta) = 0.$$

Proof of Lemma 7 and Lemma 8. Notice that the foundation for the proof of Lemma 7 and Lemma 8 has already been prepared in Lemma 5, where the local and global properties of the derivatives of the remainder term of generic dependency expansions has been analyzed. In particular, an immediate implication of Lemma 5 is that we have

$$\frac{\partial^m}{\partial \delta_{\alpha_1, \gamma_1}^{m_1} \partial \delta_{\alpha_2, \gamma_2}^{m_2} \dots \partial \delta_{\alpha_v, \gamma_v}^{m_v}} \hat{\rho}_g(0) = 0 \quad (46)$$

for any sequence of pairs $(\alpha_1, \gamma_1), \dots, (\alpha_v, \gamma_v)$ with $\alpha_k \in S_n^v$ and $\gamma_k \in \overline{\mathcal{T}}_{\alpha_k}$, which satisfies

$$\left| \bigcup_{k=1}^v \gamma_k \right| \leq v.$$

This statement is true for any sequence of integers $m_1, \dots, m_v \in \mathbb{N}$ with $\sum_{k=1}^v m_k = m$. In particular, if we have $\sum_{k=1}^v |\gamma_k| \leq v$ the above condition is satisfied and, consequently, the statement of Lemma 7 applying Taylor's theorem in combination with (46). Moreover, the statement of Lemma 8 follows from the fact that (46) holds for any order m , i.e., the error must vanish globally.³ \square

The above Lemmas motivate that a reasonable requirement on the remainder bound function $r_g(A)$ is that it inherits the local and global convergence properties of the expansion.

Requirement 1 *Let $g \in \mathcal{L}$ be any atom operation of the library \mathcal{L} and \overline{D} any compact set such that g has no singularities on \overline{D} . If the bounds $L_{\alpha, \gamma}$ and $U_{\alpha, \gamma}$ satisfy $U_{\alpha, \gamma} - L_{\alpha, \gamma} = \mathbf{O}(\text{diam}(X)^{|\gamma|})$ for all $\alpha \in S_n^v$ and all $\gamma \in \overline{\mathcal{T}}_{\alpha}$, then the remainder bound satisfies*

$$r_g(A) = \mathbf{O}(\text{diam}(X)^{v+1})$$

assuming $X \subseteq \overline{D}$.

Requirement 2 *If there exists a sequence $\alpha' \in S_n^v$ such that $L_{\alpha, \gamma} = U_{\alpha, \gamma} = 0$ for all $\alpha \in S_n^v$ and all $\gamma \in \overline{\mathcal{T}}_{\alpha}$ with $\gamma \notin \overline{\mathcal{T}}_{\alpha'}$, then the remainder bound vanishes, $r_g(A) = 0$, under the assumption that the assertion $[\lambda(A), \mu(A)] \subset D$ holds, i.e., such that g has no singularities on the considered domain.*

The procedure for constructing functions r_g that satisfy the above requirements is in principle analogous to the derivation of remainder bound formulas for the first order interval superposition arithmetic, which have been listed in Table 2. However, for the general higher order case a derivation of remainder bound formulas can become rather cumbersome, if this is done by “paper and pencil”. Fortunately, it is not necessary to work out such explicit expressions, as there exist mature tools for algebraic simplification of multivariate rational expressions [7]. Therefore, instead of deriving lengthy remainder bound formulas by hand, we propose to automate this task. If g admits a rational additional theorem this can be done by the passing through the following steps.

1. Substitute the rational addition theorem for g in order to re-write the expression (45) as a rational expression of $g(\omega)$ and $g(\delta_{\alpha, \gamma})$.
2. Use a suitable tool from the field of multivariate rational computer algebra [7] in order to simplify the rational expression.
3. Bound the simplified rational expression by applying standard interval arithmetic.

³ The conclusion that a function whose derivatives all vanish must be globally equal to 0 is formally only valid, if an analytic continuation of this function exists, which is, however, the case for all atom operations that are relevant in the context of this paper. However, the above proof can be modified by invoking symbolic counting arguments (in principle, analogous to Lemma 5). This way, it can be established that Lemma 8 also holds if no analytic continuation of g is available.

Otherwise, if g admits an inverse rational theorem, the above procedure can be applied, too, where the only difference is that g is evaluated at a rational expression of ω and $\delta_{\alpha,\gamma}$, i.e., the rational computer algebra tools have to be used to simplify the argument of g before applying standard interval arithmetic to bound the simplified or re-ordered expression. Finally, the local and global properties of higher order interval superposition arithmetic can be summarized as follows.

Theorem 7 *Let all atom operations $g \in \mathcal{L}$ be smooth on their domain and let the remainder bounds r_g of all univariate atom operations satisfy Requirement 1, let the composition rules of the proposed interval superposition arithmetic be implemented by using Algorithm 3 and 4, and let $f : D \rightarrow \mathbb{R}$ be a given factorable function over \mathcal{L} on the open domain $D \subseteq \mathbb{R}^n$. If the associated central and higher order variation points are computed by Algorithm 5, then the computed higher order interval superposition model $F_{f,X}$ satisfies*

$$\max_{x \in X} \{ \text{diam}(F_{f,X}(x)) \} = \mathbf{O} \left(\frac{\text{diam}(X)}{N} + [\text{diam}(X)]^{v+1} \right),$$

for all $X \subseteq \bar{D}$, where $\bar{D} \subset D$ is a compact domain on which f has no singularities.

Proof. The main idea of the proof of this theorem is to exploit Requirement 1 recursively in order to show that the remainder bound remains of order $\mathbf{O} \left(\frac{\text{diam}(X)}{N} + [\text{diam}(X)]^{v+1} \right)$. Here, variables can be represented with an accuracy of order $\mathbf{O} \left(\frac{\text{diam}(X)}{N} \right)$. Notice that this term can—in complete analogy to the first order interval models—be made arbitrarily small by choosing a sufficiently large N . However, due to Requirement 1 the remainder bound of the higher order interval superposition models converges with order $v + 1$. This result can be established by using the following induction argument.

1. *Induction start.* Variables can be represented with accuracy $\mathbf{O} \left(\frac{\text{diam}(X)}{N} \right)$.
2. *Induction assumption.* In the case of univariate composition, we assume that the interval superposition model of the inner model satisfies

$$\max_{x \in X} \{ \text{diam}(F_{h,X}(x)) \} = \mathbf{O} \left(\frac{\text{diam}(X)}{N} + [\text{diam}(X)]^{v+1} \right).$$

In the case of bivariate composition the same assumption needs to hold for both input models.

3. *Induction step.* The induction assumption implies

$$\text{diam}(A_\alpha^\beta) = \mathbf{O} \left(\frac{\text{diam}(X)}{N} + [\text{diam}(X)]^{v+1} \right).$$

Moreover, as all atom operations in the library \mathcal{L} are assumed to be smooth, the input function $h(x)$ is smooth, too, and consequently its interval superposition model can be approximated with accuracy

$$\max_{\beta \in \{1, \dots, N\}^v} \text{diam} \left(A_\alpha^\beta - a_\alpha - \sum_{\gamma \in \mathcal{J}_\alpha} a_{\alpha,\gamma}^{\beta_{\alpha,\gamma}} \right) = \mathbf{O} \left(\frac{\text{diam}(X)}{N} + [\text{diam}(X)]^{v+1} \right).$$

This approximation accuracy is also achieved if the central and higher order variation points are computed by Algorithm 5, as this algorithm finds the best least-squares approximation of the input model with a piecewise constant model of coupling degree

v. Moreover, in analogy to Taylor's theorem, the bounds on the higher order variation points satisfy $U_{\alpha,\gamma} - L_{\alpha,\gamma} = \mathbf{O}\left(\text{diam}(X)^{|\gamma|}\right)$ such that Requirement 1 implies that the remainder bound is accurate up to the desired order and, consequently, the composition result satisfies

$$\max_{x \in X} \{\text{diam}(F_{f,X}(x))\} = \mathbf{O}\left(\frac{\text{diam}(X)}{N} + [\text{diam}(X)]^{v+1}\right).$$

For bivariate compositions the above proof technique can be applied to both input arguments.

This proves the statement of the theorem. \square

Theorem 8 *Let all the remainder bounds r_g of all univariate atom operations $g \in \mathcal{L}$ satisfy Requirement 2 and let the composition rules of the proposed interval superposition arithmetic be implemented by using Algorithm 3 and 4. If the associated central and higher order variation points are computed by Algorithm 5 and if the factorable function f is partially separable with a coupling degree that is smaller than or equal to v , then the computed higher order interval superposition model $F_{f,X}$ satisfies*

$$\max_{x \in X} \{\text{diam}(F_{f,X}(x))\} = \mathbf{O}\left(\frac{\text{diam}(X)}{N}\right)$$

under the additional assumption that no singularity assertions are violated.

Proof. The proof of this theorem is in principle analogous to the proof of Theorem 7, although one difference that the induction argument has to be modified slightly as the local Requirement 1 has now been replaced by the global Requirement 2 and all estimates of order $\mathbf{O}\left(\frac{\text{diam}(X)}{N} + [\text{diam}(X)]^{v+1}\right)$ are replaced by estimates of order $\mathbf{O}\left(\frac{\text{diam}(X)}{N}\right)$. However, as we assume that the coupling the degree smaller than or equal to v the v -th order dependency expansion ensures that there are no terms of order $\mathbf{O}\left([\text{diam}(X)]^{v+1}\right)$, i.e., we must have

$$\max_{x \in X} \{\text{diam}(F_{f,X}(x))\} = \mathbf{O}\left(\frac{\text{diam}(X)}{N}\right),$$

as claimed by the theorem. \square

3.6 Complexity

Algorithms 3 and 4 have for any fixed order $v < \infty$ a polynomial run-time complexity. In order to be more precise the following list summarizes the storage and run-time requirements of each step of Algorithm 3 in detail. The complexity of Algorithm 4 can be analyzed analogously.

Step 1. Similar to Algorithm 1 the proposed higher order composition rule in Algorithm 3 chooses central points $a_\alpha \in \mathbb{R}$ in Step 1. Thus, in total, we need to store

$$\binom{n}{v} = \mathbf{O}\left(\frac{1}{v!} n^v\right)$$

central points. Also recall that the computational complexity of evaluating the bounds $L_\alpha(A)$ and $U_\alpha(A)$ for all $\alpha \in S_n^v$ is of order $\mathbf{O}\left(\frac{1}{v!} n^v N^v\right)$.

Step 2. In Step 2 of Algorithm 3 we have to choose higher order variation points $a_{\alpha,\gamma}^\beta \in \mathbb{R}$ for all $\beta \in \{1, \dots, N\}^{|\gamma|}$, all $\gamma \in \mathcal{S}_\alpha$ and all $\alpha \in S_n^v$. In total,

$$\binom{n}{v} \sum_{k=1}^{v-1} \binom{v}{k} N^k = \binom{n}{v} [(1+N)^v - 1 - N^v] = \mathbf{O}\left(\frac{1}{v!} n^v N^v\right)$$

higher order variation points need to be stored. Recall that these higher order variation points can always be generated by using Algorithm 5, which has a run-time complexity of order $\mathbf{O}\left(\frac{1}{v!} n^v N^v\right)$. Moreover, the run-time complexity of computing the partial sum terms ω_γ^β is given by

$$\sum_{k=1}^{v-1} \binom{n-k}{v-k} \binom{n}{k} N^k = \sum_{k=1}^{v-1} \binom{n}{v} \binom{v}{k} N^k = \mathbf{O}\left(\frac{1}{v!} n^v N^v\right)$$

This follows from the fact that the sum $\omega_\gamma^\beta = \sum_{\substack{\alpha \in S_n^v \\ \alpha \supset \gamma}} a_{\alpha,\gamma}^\beta$ has $\binom{n-k}{v-k}$ summands for every given pair (β, γ) . The cost of storing these partial sums is of the same order.

Step 3. The precise run-time for computing the bound $r_g(A)$ depends on the atom operation g and how its remainder bound is constructed. However, notice that the right-hand expression (45) has $\mathbf{O}\left(\frac{2^v}{v!} n^v N^v\right)$ terms and, thus, one might argue that an automated algebraic simplification routine should yield an expression for the remainder terms that has (at most) the same complexity, since the objective of algebraic simplification is to reduce the number of terms, not to increase them. In particular, the remainder bound expression for the product rule in Step 3 of Algorithm 4 satisfies this complexity requirement.

Step 4. For any given pair (α, β) the associated interval coefficient C_α^β can be computed by implementing $\mathbf{O}(2^v)$ sum operations, $\mathbf{O}(2^v)$ real-valued evaluations of g , and one evaluation of the interval extension of g (for the first summand). Thus, the complexity of Step 4 is of order $\mathbf{O}\left(\frac{2^v}{v!} n^v N^v\right)$.

Thus, in summary, the storage complexity of Algorithm 3 is of order $\mathbf{O}\left(\frac{1}{v!} n^v N^v\right)$ and the run-time complexity is of order $\mathbf{O}\left(\frac{2^v}{v!} n^v N^v\right)$. Notice that for any given, constant order v , this shows that Algorithm 3 has a polynomial run-time complexity. An analogous statement holds for Algorithm 4.

4 Implementation and Examples

The goal of this section is to illustrate the potential of the proposed first- and higher order interval superposition arithmetic for bounding factorable functions. For this aim, all algorithms from the previous sections have been implemented by using the programming language Julia. The purpose of this prototype implementation is to illustrate the accuracy of the proposed arithmetics and to compare it with existing enclosure algorithms for factorable functions such as standard interval arithmetics and Taylor models with interval remainders. Here, we use the library MC++ (<http://omega-icl.bitbucket.org/mcpp/>), which implements tools for the enclosing the image set of factorable functions with

Taylor models. Moreover, all results are based on interval remainder bounds. In MC++ the interval libraries PROFIL (<http://www.ti3.tu-harburg.de/>) and FILIB++ (<http://www2.math.uni-wuppertal.de/~xsc/>) are available to perform the remainder interval computations. Notice that round-off errors are not taken into account. Also notice that run-time comparison are not reported in this paper, as the results for the interval superposition arithmetics are based on a prototype implementation in Julia, which cannot be compared with the optimized C/C++ implementation of Taylor models of the mature software package MC++. The authors mention however that the scaling of the run-time with n and N as listed in the above complexity analysis section can be confirmed by the numerical experiments. A more mature C/C++ implementation of the proposed arithmetics is at planning stage and will be part of future research.

In order to numerically compare different enclosure arithmetics, the exact image set of a function $f : X \rightarrow \mathbb{R}^m$ on an n -dimensional interval domain $X \subseteq \mathbb{I}^n$ is denoted by $f(X)$. Moreover, the symbol “ $F(X)$ ” is used to denote the enclosure set $f(X) \subseteq F(X)$ that may be computed with different enclosure arithmetics. Next, the quality of the enclosure is measured by the Hausdorff distance

$$d_H(f(X), F(X)) = \max_{y \in F(X)} \min_{x \in f(X)} \|x - y\|_\infty. \quad (47)$$

Here, $\|\cdot\|_\infty$ denotes the standard ∞ -norm in \mathbb{R}^n . The following sections analyze this Hausdorff distance for different functions f and for different enclosure arithmetics.

4.1 First order interval superposition models versus Taylor models

The goal of this section is to compare the performance of first order interval superposition models versus Taylor models on wider domains. Let $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ denote a non-convex factorable function of the form

$$f(x) = \exp(\sin(x_1) + \sin(x_2)\cos(x_2))$$

on the two-dimensional domain $X = [0, \bar{x}_1] \times [0, \bar{x}_2] \subseteq \mathbb{R}^2$. Here, $\bar{x}_1 \geq 0$ and $\bar{x}_2 \geq 0$ are parameters that can be used to control the diameter of the domain X . The upper left plot in Figure 2 shows a 3-dimensional visualization of the function f on the interval domain $[0, 10] \times [0, 20]$, i.e., for $\bar{x}_1 = 10$ and $\bar{x}_2 = 20$. The upper right plot in Figure 2 shows the overestimation of five different enclosure methods for bounding f for $\bar{x}_1 = 10^{-1}$ as a function of the domain parameter $\bar{x}_2 \in [0.1, 20]$: the red solid and red dotted lines show the overestimation of Taylor models of order 1 and 2, respectively. The black solid, black dotted, and black dashed lines correspond to the overestimation of the enclosures that are obtained by using first order interval superposition models with $N = 1$, $N = 10$, and $N = 100$. Here, the overestimation is measured in terms of the Hausdorff distance (47) between the exact image set and the five different enclosure sets. The lower left plot in Figure 2 shows the overestimation of the five mentioned methods for a fixed $\bar{x}_1 = 1$ as a function of $\bar{x}_2 \in [0.1, 20]$. Similarly, the lower right plot in Figure 2 depicts the corresponding results for $\bar{x}_1 = 10$, again as a function of $\bar{x}_2 \in [0.1, 20]$. Here, the results for the Taylor models is not shown, as the overestimation error is larger than 10^7 , i.e., Taylor models do not yield reasonable enclosures on this rather large domain. In order to avoid misunderstanding at this point, notice that the width of the exact image set $f(X)$ is monotonically increasing with respect to the parameter \bar{x}_2 . However, the Hausdorff difference between $f(X)$ and an enclosure $F(X)$ is not necessarily monotonous in \bar{x}_2 . In fact, also the overestimation of standard Taylor models decreases in

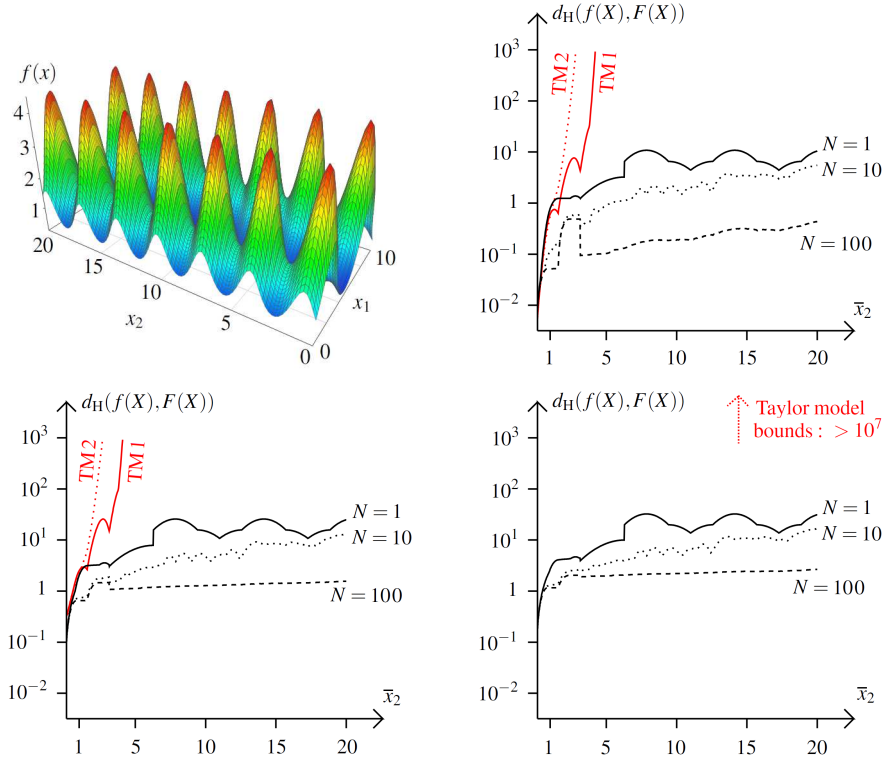


Fig. 2 Upper-left: 3-dimensional visualization of the function f on the domain $X = [0, 10] \times [0, 20]$. Upper-right: Hausdorff distance between the exact image set $f(X)$ and its enclosure sets on the interval $X = [0, 0.1] \times [0, \bar{x}_2]$ as a function of $\bar{x}_2 \in [0.1, 20]$. Lower-left: Hausdorff distance between the exact image set $f(X)$ and its enclosure sets on the interval $X = [0, 1] \times [0, \bar{x}_2]$ as a function of $\bar{x}_2 \in [0.1, 20]$. Lower-right: Hausdorff distance between the exact image set $f(X)$ and its enclosure sets on the interval $X = [0, 10] \times [0, \bar{x}_2]$ as a function of $\bar{x}_2 \in [0.1, 20]$. In all plots the black solid line corresponds to the results obtained with first order interval superposition arithmetics with $N = 1$. The black dotted lines correspond to first order interval superposition arithmetic with $N = 10$, and the black dashed lines use $N = 100$. The red solid and red dotted line correspond to the results obtained with Taylor models of order 1 and 2, respectively.

sections, if the domain X is increased, although one might argue that, overall, a rough trend is that the overestimation error of the enclosure methods increases when increasing the domain X . One aspect that is not shown in the Figure 2 is that Taylor models do outperform interval superposition models on very small domains, i.e., if we would zoom in and analyze the overestimation for $\bar{x}_1, \bar{x}_2 \leq 10^{-1}$, we could see that Taylor models are the better choice on such small domains. However, as mentioned in previous section, the local convergence properties of the proposed superposition could be improved by constructing, e.g., superpositions of first order Taylor models rather than intervals, but this is not implemented here as the motivation of this paper is to develop a tool that yields reasonable bounds on wider domains. Notice that on the domain $[0, 10] \times [0, 20]$ the overestimation of the first order interval superposition method with $N = 100$ yields an enclosure that is approximately 1.62 times larger than the width of the exact range, i.e., the relative over-approximation is approximately 62 %. This is in contrast to Taylor models, which yield bounds that are more

than 10^7 times larger than the exact image set. The performance of Taylor models of order larger than 2 is not shown in the figure, as they perform even worse than the Taylor models of order 1 and 2 on the analyzed, particularly large domains. Here again, of course, if we would zoom in on smaller domains X , we could see that increasing the Taylor model order does improve the accuracy for such smaller X , but this effect is well-known [4, 5, 39] and therefore not analyzed further at this point. Moreover, as the above example considers a function with two variables, second or other higher order interval superposition models are equivalent to exhaustive branching, i.e., this example is not suited for making a fair comparison of such higher order methods. Therefore, we refer to Section 4.2, where first and second order interval superposition models are analyzed for a more challenging case study.

4.2 Performance of second order interval superposition models

In order to illustrate how the proposed interval superposition arithmetics performs for a more challenging example, we introduce the function

$$f_1(x) = \begin{pmatrix} -x_1^2 + x_1^4 + \frac{4}{10}x_1x_3 + \frac{1}{10}x_2^2 \\ \frac{1}{4}x_2 + \frac{1}{4}x_2x_3 \\ -x_3^2 + x_3^4 + \frac{4}{10}x_1x_3 + \frac{1}{10}x_2^2 \end{pmatrix}. \quad (48)$$

Notice that f_1 is a multivariate polynomial of order 4. In the next step we define the functions

$$\forall k \in \mathbb{N}, \quad f_{k+1}(x) = f_1(f_k(x)). \quad (49)$$

recursively. Of course, the function f_k are “simple” polynomial functions in principle, but the actual challenge is that the order of these polynomials grows exponentially with k . More precisely, f_k is a polynomial function of order 4^k . The goal of this section is to find enclosure sets of the exact image sets $f_k(X)$ on the rather large interval domain $X = [-1, 1]^3 \subseteq \mathbb{R}^3$. It can be checked easily that exact image sets satisfy a convergence rate condition of the form

$$\lim_{k \rightarrow \infty} \frac{d_H(\{0\}, f_k(X))}{4^k} = 1,$$

i.e., the exact image set contracts to 0 with linear convergence rate, $\text{diam}(f_k(X)) \approx 4^{-k}$ for sufficiently large k . Figure 3 shows the Hausdorff distance between the exact image set $f_k(X)$ and the enclosure sets that are obtained by applying standard interval arithmetic, first order interval superposition arithmetic with $N = 100$, as well as second order interval superposition arithmetic also with $N = 100$. All results are shown in dependence on k . Notice that the first order interval superposition models yield enclosure sets that are much less conservative than the enclosures that are obtained by standard interval arithmetic, but both enclosure methods are divergent for $k \rightarrow \infty$. This is in contrast to the second order interval superposition arithmetic, which yields convergent enclosures for $k \rightarrow \infty$. The results for first and second order Taylor models are not shown in the figure, as they perform even worse than standard interval arithmetic for this particular example. If we use a Taylor model of order 4, the remainder interval is equal to 0 when bounding the 4-th order polynomial f_1 . However, the enclosure set that is obtained by applying a 4-th order Taylor model for bounding the function f_2 is again worse than the corresponding enclosure obtained by standard interval calculus. The authors also tried to use a Taylor model of order 16, which does lead to a remainder interval of width 0 when applied to the functions f_1 and f_2 recalling that f_2 is a

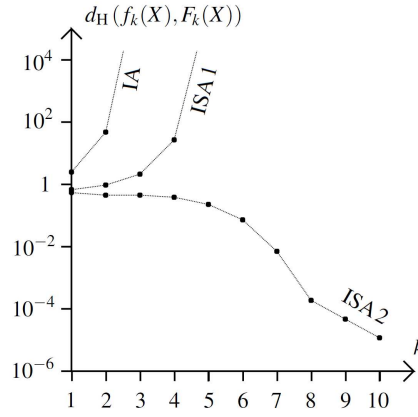


Fig. 3 The Hausdorff distance between the exact image sets $f_k(X)$ and their computed enclosure sets $F_k(X)$ in dependence on the running index k . The corresponding results for standard interval arithmetic are labeled as “IA”. The other results are obtained by using interval superposition models of order 1 and 2 with $N = 100$ labeled as “ISA1” and “ISA2”, respectively.

multivariate polynomial of order 16. However, if we apply a Taylor of order 16 for bounding the function f_3 the overestimation error is again larger than 10^6 , which corresponds to about the same accuracy that is obtained by applying the much cheaper standard interval calculus. Thus, in summary, Taylor models fail to yield reasonable bounds for this example on the particularly wide domain $X = [-1, 1]^3$. Of course, this conclusion changes if the functions f_k are bounded on smaller interval domains, where Taylor models tend to yield more accurate enclosures.

Because the exact image sets converge linearly and because Theorem 7 predicts that the overestimation error of second order interval superpositions converge has the form

$$\max_{x \in Y} \{ \text{diam}(F_{f,Y}(x)) \} = \mathbf{O} \left(\frac{\text{diam}(Y)}{N} + [\text{diam}(Y)]^3 \right),$$

with respect to the diameter of the input set Y , one would expect that the overestimation error converges cubically for larger input domains Y and then linearly, once term the linearly convergent term becomes dominant. The convergence plot of the second order interval superposition models in Figure 3 indicates indeed such a behavior: for the range $k \in \{5, 6, 7, 8\}$ the Hausdorff distance between the second order interval superposition enclosure and the exact set converges superlinearly, although one might argue that it is hard to tell from these numerical results whether this range corresponds to a cubic convergence phase. However, for $k \geq 8$ the convergence rate switches and becomes linear, since in this phase the terms with linear convergence order, $\mathbf{O}(N^{-1} \text{diam}(Y))$, dominate recalling that these terms did not dominate early because we chose a rather large number of branches, $N = 100$, in this example. Consequently, the numerical convergence behavior of the second order interval superposition models for $k \geq 5$ can be explained by the theoretical analysis results from Theorem 7.

5 Conclusions

This paper has introduced an interval superposition arithmetic and illustrated the advantages of this new arithmetic compared to existing enclosure methods for factorable functions on wider domains. The construction of interval superposition models is based on derivative-free composition rules which exploit global algebraic properties such as rational addition theorems, inverse rational addition theorems, and partially separable sub-structures of the computational graph of factorable functions. The corresponding technical developments are based on dependency expansions, which can be viewed as a derivative-free, algebraic abstraction of Taylor expansions. Interval superposition arithmetic has polynomial run-time and storage complexity of order $\mathcal{O}(n^v N^v)$, which depends on the number n of variables of the factorable functions and the branching accuracy N . The order v of this polynomial run-time bound is equal to the order of the dependency expansion. Moreover, this paper has established local and global convergence estimates of the proposed arithmetic in dependence on the order v of the interval superposition and in dependence on the coupling degree of the given factorable function. From a practical perspective, the main advantage of interval superposition arithmetics compared to other enclosure methods is that it yields reasonably accurate bounds of the image set of factorable functions on wider interval domains, for which existing methods often yield divergent or very conservative bounds. This advantage has been illustrated by analyzing two numerical case studies.

Many algorithms of current interest such as semi-infinite programming algorithms, validated integrators, as well as Branch & Bound methods for global optimization rely on the availability of set arithmetics for factorable functions. As interval superposition models have major advantages compared to existing set computation methods on wider domains, this novel arithmetic should be of practical relevance to everyone working on or using such algorithms. Moreover, since interval superposition arithmetic is, on the one hand, a bounding tool, but, on the other hand, exploits the concept of coordinate aligned branching while enforcing polynomial run-time and storage requirements, this arithmetic might lead to new types of global optimization algorithms, where the branching and bounding operations are not considered as separate routines anymore. However, a deeper analysis of the corresponding interplay between the presence of global algebraic structures in factorable functions and the complexity of their associated optimization problems is beyond the scope of this paper and shall be part of future research.

References

1. Y. Abe. A statement of Weierstrass on meromorphic functions which admit an algebraic addition theorem. *J. Math. Soc. Japan* 57(3):709–723, 2005.
2. S.A. Andrea. Algebraic addition theorems. *Adv. Math.* 13:20–30, 1974.
3. Z. Battles, L.N. Trefethen. An extension of MATLAB to continuous functions and operators. *SIAM J. Sci. Comput.* 25:1743–1770, 2004.
4. M. Berz. From Taylor series to Taylor models. In *Nonlinear Problems in Accelerator Physics*, American Institute of Physics CP405, pp.:1–27, 1997.
5. M. Berz, G. Hoffstätter. Computation and application of Taylor polynomials with remainder bounds. *Reliab. Comput.* 4:83–97, 1998.
6. A. Bompadre, A. Mitsos, B. Chachuat. Convergence analysis of Taylor and McCormick-Taylor models. *Journal of Global Optimization* 57(1):75–114, 2013.

7. B. Buchberger, R. Loos. Algebraic Simplification. In: Computer Algebra - Symbolic and Algebraic Computation. B. Buchberger, G. E. Collins, R. Loos (eds.), pp. 11–43, 1982.
8. B. Chachuat, B. Houska, R. Paulen, N. Peric, J. Rajyaguru, M.E. Villanueva. Set theoretic approaches in analysis, estimation and control of nonlinear systems. IFAC-PapersOnLine Volume 48(8), pp:981–995, 2015.
9. A.R. Conn, N.I.M. Gould, Ph.L. Toint. An introduction to the structure of large scale nonlinear optimization problems and the LANCELOT project. In: Glowinski, R., Lichnewsky, A. (eds.) Computing Methods in Applied Sciences and Engineering, pp. 4251. SIAM, Philadelphia, 1990.
10. Conn, A.R., Gould, N.I.M., Toint, Ph.L.: Improving the decomposition of partially separable functions in the context of large-scale optimization: a first approach. In: W.W. Hager, D.W. Hearn, P.M. Pardalos (eds.). Large Scale Optimization: State of the Art, pp. 82–94. Kluwer Academic Publishers, Amsterdam, 1994.
11. K.S. Du, R.B. Kearfott. The cluster problem in multivariate global optimization. *Journal of Global Optimization* 5(3):253–265, 1994.
12. J.P. Eckmann, H. Koch, P. Wittwer. A computer-assisted proof of universality in area-preserving maps. *Memoirs of the AMS* 47:289, 1984.
13. L.H. de Figueiredo, J. Stolfi. Affine arithmetic: Concepts and applications. *Numerical Algorithms* 37(1-4):147–158, 2004.
14. C.A. Floudas and O. Stein. The Adaptative Convexification Algorithm: a Feasible Point Method for Semi-Infinite Programming. *SIAM Journal on Optimization*, 18(4):1187–1208, 2007.
15. C.A. Floudas. Deterministic global optimization: theory, methods and applications. Springer Science & Business Media, Vol. 37, 2013.
16. A. Griewank and A. Walther. *Evaluating Derivatives*. SIAM, 2008.
17. M.X. Goemans and D.P. Williamson. Improved approximation algorithms for Maximum Cut and satisfiability problems using semidefinite programming. *Journal of ACM*, 42:1115–1145, 1995.
18. D. Henrion, S. Tarbouriech, and D. Arzelier. LMI Approximations for the Radius of the Intersection of Ellipsoids: A Survey. *Journal of Optimization Theory and Applications*, 108(1):1–28, 2001.
19. B. Houska, F. Logist, J. Van Impe, M. Diehl. Robust Optimization of Nonlinear Dynamic Systems with Application to a Jacketed Tubular Reactor. *Journal of Process Control*, Volume 22(6), pp. 1152–1160, 2012.
20. B. Houska, M.E. Villanueva, B. Chachuat. Stable Set-Valued Integration of Nonlinear Dynamic Systems using Affine Set Parameterizations. *SIAM Journal on Numerical Analysis*, 53(5), pp:2307–2328, 2015.
21. A. Kurzhanski, I. Valyi. Ellipsoidal Calculus for Estimation and Control. Series in Systems & Control: Foundations & Applications, Birkhäuser, 1997.
22. J.B. Lasserre. Moments, Positive Polynomials and Their Applications. Imperial College Press, 2009.
23. Q. Lin, J.G. Rokne. Methods for bounding the range of a polynomial. *J. Comput Appl Math* 58:193–199, 1995.
24. M. Neher, K.R. Jackson, N.S. Nedialkov. On Taylor model based integration of ODEs. *SIAM Journal on Numerical Analysis* 45:236–262, 2007.
25. A. Nemirovski, C. Roos, and T. Terlaky. On maximization of quadratic form over intersection of ellipsoids with common center. *Mathematical Programming*, 86(3):463–473, 1999.

26. Y. Nesterov. Semidefinite relaxation and non-convex quadratic optimization. *Optimization Methods and Software*, 12:1–20, 1997.
27. A. Neumaier. Complete search in continuous global optimization and constraint satisfaction. *Acta Numer.* 13:271–369, 2004.
28. K. Makino, M. Berz. Efficient control of the dependency problem based on Taylor model methods. *Reliab. Comput.* 5(1):3–12, 1999.
29. G.P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I – Convex underestimating problems. *Mathematical Programming* 10:147–175, 1976.
30. R. Misener, C.A. Floudas. GloMIQO: Global Mixed-Integer Quadratic Optimizer. *Journal of Global Optimization*, 57(1):3–50, 2013.
31. A. Mitsos, P. Lemonidis, P.I. Barton. Global solution of bilevel programs with a non-convex inner program. *Journal of Global Optimization* 42.4:475–513, 2008.
32. A. Mitsos, B. Chachuat, P.I. Barton. McCormick-based relaxations of algorithms. *SIAM Journal on Optimization* 20(2):573–601, 2009.
33. R.E. Moore. *Interval Analysis*. Prentice-Hall, Englewood Cliffs, NJ, 2966.
34. R.E. Moore, R.B. Kearfott, M.J. Cloud. *Introduction to Interval Analysis*. SIAM, Philadelphia, PA, 2009.
35. P.M. Pardalos, S.A. Vavasis. Quadratic programming with one negative eigenvalue is NP-hard. *Journal of Global Optimization* 1 (1): 1522, 1991.
36. J. Rajyaguru, M.E. Villanueva, B. Houska, B. Chachuat. Higher-Order Inclusions of Factorable Functions by Chebyshev Models. *Journal of Global Optimization*, 2016. (submitted)
37. H. Ratschek, J. Rokne. *Computer Methods for the Range of Functions*. Series in Mathematics and Its Applications, Ellis Horwood Ltd, Mathematics and Its Applications, Chichester, UK, 1984.
38. N.V. Sahinidis. A general purpose global optimization software package. *Journal of Global Optimization*, 8(2):201–205, 1996.
39. A.M. Sahlodin, B. Chachuat. Convex/concave relaxations of parametric ODEs using Taylor models. *Computers and Chemical Engineering* 35(5):844–857, 2011.
40. M. Tawarmalani, N.V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103(2):225–249, 2005.
41. L.N. Trefethen. Computing numerically with functions instead of numbers. *Math. Comput. Sci.* 1:9–19, 2007.
42. A. Townsend, L.N. Trefethen. An extension of Chebfun to two dimensions. *SIAM J. Sci. Comput* 35(6):C495–C498, 2013.
43. M.E. Villanueva, J. Rajyaguru, B. Houska, B. Chachuat. Ellipsoidal arithmetic for multivariate systems. *Comput. Aided Chem. Eng.* 37:767–772, 2015.
44. M.E. Villanueva, B. Houska, B. Chachuat. Unified Framework for the Propagation of Continuous-Time Enclosures for Parametric Nonlinear ODEs. *J. of Global Optim* 62(3), pp:575–613, 2015.
45. M.B. Villarino. Algebraic Addition Theorems. Retrieved February 12, 2016, from the arXiv database, arXiv:1212.6471, 2013.
46. A. Wechsung, S.D. Schaber, P.I. Barton. The cluster problem revisited. *Journal of Global Optimization* 58(3):429–438, 2014.

A Derivation of the remainder bounds from Table 2

This section briefly discusses how to derive remainder bounds for the proposed first order interval superposition arithmetic. These remainder bounds are needed in the composition rule of the first order interval superposition arithmetic (Algorithm 1) requiring that the inequality

$$\left| \sum_{i=1}^n g(\omega + \delta_i) - (n-1)g(\omega) - g\left(\omega + \sum_{i=1}^n \delta_i\right) \right| \leq r_g(A) \quad (50)$$

is satisfied for all $\delta \in \mathbb{R}^n$ with $\forall i \in \{1, \dots, n\}$, $L(A_i) \leq a_i + \delta_i \in U(A_i)$. In the sections below we verify this inequality for all rows of Table 2 except for the remainder bounds for the univariate minus and univariate square in the first two rows in Table 2, which are easy to check and left as an exercise for the reader.

A.1 Exponential

For the atom function $g(x) = e^x$ we have to bound the expression

$$\sum_{i=1}^n g(\omega + \delta_i) - (n-1)g(\omega) - g\left(\omega + \sum_{i=1}^n \delta_i\right) = e^\omega \left[\sum_{i=1}^n e^{\delta_i} - (n-1) - \prod_{i=1}^n e^{\delta_i} \right]$$

for all δ_i with $L(A_i) \leq a_i + \delta_i \leq U(A_i)$. In the latter equation, we follow the concept of Strategy 1; that is, we apply the addition theorem for the exponential function,

$$e^{\omega + \delta_i} = e^\omega e^{\delta_i} \quad \text{and} \quad e^{\omega + \sum_{i=1}^n \delta_i} = e^\omega \prod_{i=1}^n e^{\delta_i}.$$

It is convenient to introduce the auxiliary variables $t_i = e^{\delta_i} - 1$ such that

$$e^\omega \left[\sum_{i=1}^n e^{\delta_i} - (n-1) - \prod_{i=1}^n e^{\delta_i} \right] = e^\omega \left[\sum_{i=1}^n t_i + 1 - \prod_{i=1}^n (1 + t_i) \right]. \quad (51)$$

The absolute value of this expression can be bounded as

$$e^\omega \left| \sum_{i=1}^n t_i + 1 - \prod_{i=1}^n (1 + t_i) \right| \leq e^\omega \left(\prod_{i=1}^n (1 + s_i) - \sum_{i=1}^n s_i - 1 \right) \quad \text{with} \quad s_i = \max \left\{ e^{U(A_i) - a_i} - 1, 1 - e^{L(A_i) - a_i} \right\}.$$

This motivates to choose the central points $a_i = \log \left(\frac{1}{2} (e^{U(A_i)} + e^{L(A_i)}) \right)$ such that s_i takes the smallest possible value, given by

$$s_i = \frac{e^{U(A_i)} - e^{L(A_i)}}{e^{U(A_i)} + e^{L(A_i)}}.$$

In summary, we have shown that

$$\left| \sum_{i=1}^n g(\omega + \delta_i) - (n-1)g(\omega) - g\left(\omega + \sum_{i=1}^n \delta_i\right) \right| \leq e^\omega \left(\prod_{i=1}^n (1 + s_i) - \sum_{i=1}^n s_i - 1 \right) = r_g(A).$$

This is the remainder bound from that is listed in the fourth row of Table 2.

A.2 Inverse

The aim of this section is to find a remainder bound for the atom function $g(x) = \frac{1}{x}$ on the positive domain $\mathbb{R}_{++} = \{x \mid x > 0\}$. This is sufficient, as bounds on the domain $\mathbb{R}_{--} = \{x \mid x < 0\}$ can be found analogously. If an interval contains 0, the bounds are set to $[-\infty, \infty]$. The main idea is to exploit the addition theorem for

the inverse function and to apply further standard manipulations for rational functions in order to simplify the result. We start with the equation

$$\begin{aligned} \sum_{i=1}^n g(\omega + \delta_i) - (n-1)g(\omega) - g\left(\omega + \sum_{i=1}^n \delta_i\right) &= \sum_{i=1}^n \frac{1}{\omega + \delta_i} - \frac{1}{\omega + \sum_{i=1}^n \delta_i} - \frac{n-1}{\omega} \\ &= \frac{1}{\omega} \left(\sum_{i=1}^n \frac{-\delta_i}{\omega + \delta_i} + \frac{\sum_{i=1}^n \delta_i}{\omega + \sum_{i=1}^n \delta_i} \right) \\ &= \frac{1}{\omega} \frac{1}{\omega + \sum_{i=1}^n \delta_i} \left(\sum_{i=1}^n \frac{\delta_i(\delta_i - \sum_{k=1}^n \delta_k)}{\omega + \delta_i} \right). \end{aligned}$$

Next, we bound the terms in the last equation separately under the assumption that $\lambda(A) > 0$ (if we don't have $\lambda(A) > 0$, we cannot ensure that 0 is not in the interval),

$$\begin{aligned} \left| \frac{1}{\omega + \sum_{i=1}^n \delta_i} \right| &\leq \frac{1}{\lambda(A)}, \\ \left| \frac{\delta_i}{\omega + \delta_i} \right| &\leq \max \left\{ \frac{a_i - L(A_i)}{\omega - a_i + L(A_i)}, \frac{U(A_i) - a_i}{\omega - a_i + U(A_i)} \right\} = s_i, \\ \text{and } \left| \delta_i - \sum_{k=1}^n \delta_k \right| &\leq \mu(A) - \omega - (U(A_i) - a_i). \end{aligned}$$

Substituting these inequalities yields

$$\left| \sum_{i=1}^n g(\omega + \delta_i) - (n-1)g(\omega) - g\left(\omega + \sum_{i=1}^n \delta_i\right) \right| \leq \frac{1}{\omega \lambda(A)} \sum_{i=1}^n s_i (\mu(A) - \omega - (U(A_i) - a_i)) = r_g(A).$$

This is the remainder bound that is listed in the third row of Table 2.

A.3 Logarithm

The aim of this section is to find a remainder bound for the atom function $g(x) = \log(x)$ on the positive domain $\mathbb{R}_{++} = \{x \mid x > 0\}$. We start by exploiting the inverse addition theorem for the logarithm, which leads to the equation

$$\begin{aligned} \sum_{i=1}^n g(\omega + \delta_i) - (n-1)g(\omega) - g\left(\omega + \sum_{i=1}^n \delta_i\right) &= \sum_{i=1}^n \log(\omega + \delta_i) - \log\left(\omega + \sum_{i=1}^n \delta_i\right) - (n-1)\log(\omega) \\ &= \log\left(\frac{\prod_{i=1}^n (\omega + \delta_i)}{\omega^{n-1}(\omega + \sum_{i=1}^n \delta_i)}\right) \\ &= \log\left(1 + \frac{\prod_{i=1}^n (\omega + \delta_i) - \omega^{n-1}(\omega + \sum_{i=1}^n \delta_i)}{\omega^{n-1}(\omega + \sum_{i=1}^n \delta_i)}\right). \end{aligned}$$

Next we bound the absolute value of this term by choosing the central points $a_i = \frac{U(A_i) + L(A_i)}{2}$ such that $|\delta_i| \leq s_i = \frac{U(A_i) - L(A_i)}{2}$ and

$$\left| \sum_{i=1}^n g(\omega + \delta_i) - (n-1)g(\omega) - g\left(\omega + \sum_{i=1}^n \delta_i\right) \right| \leq -\log\left(1 - \frac{\prod_{i=1}^n (\omega + s_i) - \omega^{n-1}(\omega + \sum_{i=1}^n s_i)}{\omega^{n-1} \lambda(A)}\right) = r_g(A).$$

This is the remainder bound that is listed in the fifth row of Table 2.

A.4 Sine and Cosine

The real-valued sine and cosine functions are similar to the exponential function in the sense that an addition theorem is available and can be used for constructing remainder terms. In order to exploit this relation systematically, we use Euler's formula, $e^{ix} = \cos(x) + i \sin(x)$ with $i = \sqrt{-1}$. The derivation passes through the following steps.

Step 1 In the first step, we derive for all $k \in \{1, \dots, n\}$ the bound

$$\begin{aligned} |e^{\pm i\delta_k} - 1| &= |\cos(\pm\delta_k) - 1 + i \sin(\pm\delta_k)| \\ &= 2 \left| \sin\left(\pm \frac{\delta_k}{2}\right) \right| \\ &\leq 2 \left| \sin\left(\left[-\frac{U(A_k) - L(A_k)}{4}, \frac{U(A_k) - L(A_k)}{4}\right]\right) \right| = s_k. \end{aligned} \quad (52)$$

Here, the expression for the scalars s_k is evaluated by using standard interval arithmetic, i.e., we define

$$s_k = 2 \left| \sin\left(\left[-\frac{U(A_k) - L(A_k)}{4}, \frac{U(A_k) - L(A_k)}{4}\right]\right) \right| = \begin{cases} 2 \sin\left(\frac{U(A_k) - L(A_k)}{4}\right) & \text{if } \frac{U(A_k) - L(A_k)}{4} \leq \frac{\pi}{2} \\ 2 & \text{otherwise} \end{cases}.$$

Step 2 In the second step, we use the bounds s_k to derive the auxiliary inequalities

$$\left| \sum_{k=1}^n e^{\pm i\delta_k} - \prod_{k=1}^n e^{\pm i\delta_k} - (n-1) \right| \leq \prod_{k=1}^n (1 + s_k) - \sum_{k=1}^n s_k - 1.$$

Step 3 The auxiliary inequalities from Step 2, in turn, can be used to establish the inequalities

$$\begin{aligned} \left| \sum_{k=1}^n \cos(\delta_k) - \cos\left(\sum_{k=1}^n \delta_k\right) - (n-1) \right| &= \frac{1}{2} \left| \sum_{k=1}^n e^{i\delta_k} + \sum_{k=1}^n e^{-i\delta_k} - \prod_{k=1}^n e^{i\delta_k} - \prod_{k=1}^n e^{-i\delta_k} - 2(n-1) \right| \\ &\leq \prod_{k=1}^n (1 + s_k) - \sum_{k=1}^n s_k - 1 \end{aligned} \quad (53)$$

as well as

$$\left| \sum_{k=1}^n \sin(\delta_k) - \sin\left(\sum_{k=1}^n \delta_k\right) \right| \leq \prod_{k=1}^n (1 + s_k) - \sum_{k=1}^n s_k - 1$$

using an analogous argument.

Step 4 For the sine function $g(x) = \sin(x)$, the estimate from Step 3 can be used to find the remainder bound

$$\begin{aligned} R_{\sin}(\delta) &= \left| \sum_{k=1}^n g(\omega + \delta_k) - (n-1)g(\omega) - g\left(\omega + \sum_{k=1}^n \delta_k\right) \right| \\ &= \left| \sum_{k=1}^n \sin(\omega + \delta_k) - \sin\left(\omega + \sum_{k=1}^n \delta_k\right) - (n-1)\sin(\omega) \right| \\ &= \left| \sin(\omega) \left(\sum_{k=1}^n \cos(\delta_k) - \cos\left(\sum_{k=1}^n \delta_k\right) - (n-1) \right) + \cos(\omega) \left(\sum_{k=1}^n \sin(\delta_k) - \sin\left(\sum_{k=1}^n \delta_k\right) \right) \right| \\ &\leq (|\sin(\omega)| + |\cos(\omega)|) \left(\prod_{k=1}^n (1 + s_k) - \sum_{k=1}^n s_k - 1 \right) = r_g(A). \end{aligned}$$

The corresponding bound for the atom function $g(x) = \cos(x)$ is given by

$$\begin{aligned}
 R_{\cos}(\delta) &= \left| \sum_{k=1}^n g(\omega + \delta_k) - (n-1)g(\omega) - g\left(\omega + \sum_{k=1}^n \delta_k\right) \right| \\
 &= \left| \sum_{k=1}^n \cos(\omega + \delta_k) - \cos\left(\omega + \sum_{k=1}^n \delta_k\right) - (n-1)\cos(\omega) \right| \\
 &= \left| \cos(\omega) \left(\sum_{k=1}^n \cos(\delta_k) - \cos\left(\sum_{k=1}^n \delta_k\right) - (n-1) \right) - \sin(\omega) \left(\sum_{k=1}^n \sin(\delta_k) - \sin\left(\sum_{k=1}^n \delta_k\right) \right) \right| \\
 &\leq (|\sin(\omega)| + |\cos(\omega)|) \left(\prod_{k=1}^n (1 + s_k) - \sum_{k=1}^n s_k - 1 \right) = r_g(A) .
 \end{aligned}$$

The above remainder bounds are listed in Table 2.

A.5 Tangent

In order to construct a remainder bound for the function $g(x) = \tan(x)$ on the open domain $(-\frac{\pi}{2}, \frac{\pi}{2})$ it is helpful to notice that the addition theorem for the tangent function, given by

$$\tan(x+y) = \frac{\tan(x) + \tan(y)}{1 - \tan(x)\tan(y)} ,$$

can alternatively be written in the difference form

$$\tan(x+y) - \tan(x) - \tan(y) = \tan(x)\tan(y)\tan(x+y) . \quad (54)$$

The correctness of this equation can be verified by multiplying the addition theorem for the tangent function by $1 - \tan(x)\tan(y)$ on both sides and by re-bracketing terms. Next, a generalization of the difference formula (54) for general sums is given by the equation

$$\rho_0(\delta) = \tan\left(\sum_{i=1}^n \delta_i\right) - \sum_{i=1}^n \tan(\delta_i) = \sum_{i=1}^{n-1} \tan(\delta_{i+1}) \tan\left(\sum_{k=1}^i \delta_k\right) \tan\left(\sum_{k=1}^{i+1} \delta_k\right) \quad (55)$$

This equation can be proven by induction. Firstly, (55) is true for $n=2$, as this case reduces to (54). Secondly, if (55) is true for a given n , we have

$$\begin{aligned}
 \tan\left(\sum_{i=1}^{n+1} \delta_i\right) - \sum_{i=1}^{n+1} \tan(\delta_i) &= \tan\left(\sum_{i=1}^{n+1} \delta_i\right) - \tan\left(\sum_{i=1}^n \delta_i\right) - \tan(\delta_{n+1}) + \tan\left(\sum_{i=1}^n \delta_i\right) - \sum_{i=1}^n \tan(\delta_i) \\
 &\stackrel{(54)}{=} \tan\left(\sum_{i=1}^{n+1} \delta_i\right) \tan\left(\sum_{i=1}^n \delta_i\right) \tan(\delta_{n+1}) + \left[\tan\left(\sum_{i=1}^n \delta_i\right) - \sum_{i=1}^n \tan(\delta_i) \right] \\
 &\stackrel{(55)}{=} \sum_{i=1}^n \tan(\delta_{i+1}) \tan\left(\sum_{k=1}^i \delta_k\right) \tan\left(\sum_{k=1}^{i+1} \delta_k\right) , \quad (56)
 \end{aligned}$$

which completes the induction step. Thus, the difference formula (55) holds for all integers n . In order to generalize the above formula further for the case $\omega \neq 0$, the following algebraic manipulations are made

$$\begin{aligned}
\rho(\delta) &= g\left(\omega + \sum_{i=1}^n \delta_i\right) + (n-1)g(\omega) - \sum_{i=1}^n g(\omega + \delta_i) \\
&= \left[\tan\left(\omega + \sum_{i=1}^n \delta_i\right) - \tan(\omega) \right] - \sum_{i=1}^n [\tan(\omega + \delta_i) - \tan(\omega)] \\
&\stackrel{(54)}{=} \tan\left(\sum_{i=1}^n \delta_i\right) \left[1 + \tan\left(\omega + \sum_{i=1}^n \delta_i\right) \tan(\omega) \right] - \sum_{i=1}^n \tan(\delta_i) [1 + \tan(\omega + \delta_i) \tan(\omega)] \\
&= \left(\tan\left(\sum_{i=1}^n \delta_i\right) - \sum_{i=1}^n \tan(\delta_i) \right) + \tan(\omega) \left(\tan\left(\sum_{i=1}^n \delta_i\right) \tan\left(\omega + \sum_{i=1}^n \delta_i\right) - \sum_{i=1}^n \tan(\delta_i) \tan(\omega + \delta_i) \right) \\
&\stackrel{(55)}{=} \rho_0(\delta) + \tan(\omega) \left(\rho_0(\delta) \tan\left(\omega + \sum_{i=1}^n \delta_i\right) + \sum_{i=1}^n \tan(\delta_i) \left[\tan\left(\omega + \sum_{i=1}^n \delta_i\right) - \tan(\omega + \delta_i) \right] \right) \\
&= \rho_0(\delta) \left[1 + \tan(\omega) \tan\left(\omega + \sum_{i=1}^n \delta_i\right) \right] \\
&\quad + \sum_{i=1}^n \tan(\omega) \tan(\delta_i) \tan\left(\sum_{k \neq i}^n \delta_k\right) \left[1 + \tan(\omega + \delta_i) \tan\left(\sum_{k \neq i}^n \delta_k\right) \tan\left(\omega + \sum_{i=1}^n \delta_i\right) \right].
\end{aligned}$$

Now, we can bound the right-hand expression by using standard interval arithmetic. This leads to a remainder bound of the form

$$\begin{aligned}
\rho(\delta) \leq r_g(A) &= \left| \sum_{i=1}^{n-1} \tan(S_{i+1}) \tan\left(\sum_{k=1}^i S_k\right) \tan\left(\sum_{k=1}^{i+1} S_k\right) [1 + \tan(\omega) \tan(\omega + \Sigma)] \right. \\
&\quad \left. + \sum_{i=1}^n \tan(\omega) \tan(S_i) \tan(T_i) [1 + \tan(\omega + S_i) \tan(T_i) \tan(\omega + \Sigma)] \right|,
\end{aligned}$$

where we have introduced the auxiliary variables

$$s_i = \frac{U(A_i) - L(A_i)}{2}, \quad S_i = [-s_i, s_i] \quad \text{and} \quad \sigma = \sum_{i=1}^n s_i, \quad \Sigma = [-\sigma, \sigma], \quad T_i = [-\sigma + s_i, \sigma - s_i].$$

This is the remainder bound for the tangent function that is listed in the last row of Table 2.